VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2017 Bc. Martin Lukačovič

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

SEGMENTACE OBRAZU S VYUŽITÍM HLUBOKÉHO UČENÍ
IMAGE SEGMENTATION USING DEEPLEARNING METHODS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Martin Lukačovič**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Jan Mašek, Ph.D.**
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Martin Lukačovič ***ID:*** 146894
Ročník: 2 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Segmentace obrazu s využitím hlubokého učení

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte metody pro segmentaci obrazu s využitím hlubokého učení a s pomocí volně dostupného prostředí vytvořte systém pro trénování a testování modelů pro segmentaci. V rámci práce bude vytvořeno uživatelské rozhraní umožňující snadné trénování a testování modelu. Dále budou vytvořeny vlastní databáze a natrénovány vlastní modely. Přesnost bude ověřena na příkladech a výsledky zobrazeny v grafech.

DOPORUČENÁ LITERATURA:

[1] S. Zheng et al., "Conditional Random Fields as Recurrent Neural Networks," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1529-1537, doi: 10.1109/ICCV.2015.179

[2] YANGQING JIA, EVAN SHELHAMER, JEFF DONAHUE, SERGEY KARAYEV, JONATHAN LONG, ROSS GIRSHICK, SERGIO GUADARRAMA a TREVOR DARRELL. Caffe. In: Proceedings of the ACM International Conference on Multimedia - MM '14 [online]. 2014 [cit. 2015-10-19]. DOI: 10.1145/2647868.2654889.

Termín zadání: 1.2.2017 ***Termín odevzdání:*** 24.5.2017

Vedoucí práce: Ing. Jan Mašek, Ph.D.
Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Práca sa zaoberá súčasnými metódami segmentácie obrazu s využitím hlbokého učenia. Opísané sú taktiež ostatné prístupy s uplatnením neurónových sietí v oblasti hlbokého učenia. Obsahuje historické riešenia neurónových sietí, ich vývoj a základný princíp. Konvolučné neurónové siete majú v súčasnosti najväčšie využitie pri riešení úloh ako je detekcia, klasifikácia a segmentácia obrazu. Pre overenie funkčnosti bolo v práci použité voľne dostupné prostredie na princípe podmienených náhodných polí ako rekurentných neurónových sietí a jeho porovnanie s využitím hlbokých konvolučných neurónových sietí s dodatočnou aplikáciou podmienených náhodných polí. Posledná zmienená metóda sa stala základom pre trénovanie vlastného modelu na dvoch rôznych datasetoch. Za účelom implementácie neurónových sietí s využitím hlbokého učenia existujú rôzne prostredia, ktoré ponúkajú rozmanité možnosti prevedenia. Pre demonštračné účely bolo v práci navrhnuté webové rozhranie v jazyku Python a pre realizáciu segmentácie bolo zvolené prostredie BVLC / Caffe. Najlepšia dosiahnutá presnosť vlastného trénovaného modelu pre segmentáciu oblečenia je 50,74 % a pre segmentáciu VOC objektov je to 68,52 %. Webové rozhranie sa uplatňuje ako nástroj pre jednoduchú interakciu užívateľa s cieľom segmentácie obrazu trénovanými modelmi.

KĽÚČOVÉ SLOVÁ

BVLC / Caffe, CNN, hlboké učenie, neurónové siete, podmienené náhodné polia, segmentácia oblečenia, segmentácia obrazu

ABSTRACT

This thesis deals with the current methods of semantic segmentation using deep learning. Other approaches of neaural networks in the area of deep learning are also discussed. It contains historical solutions of neural networks, their development, and basic principle. Convolutional neural networks are nowadays the most preferable networks in solving tasks as detection, classification, and image segmentation. The functionality was verified on a freely available environment based on conditional random fields as recurrent neural networks and compered with the deep convolutional neural networks using conditional random fields as postprocess. The latter mentioned method has become the basis for training of new models on two different datasets. There are various enviroments used to implement neural networks using deep learning, which offer diverse perform possibilities. For demonstration purposes a Python application leveraging the BVLC / Caffe framework was created. The best achieved accuracy of a trained model for clothing segmentation is 50,74 % and 68,52 % for segmentation of VOC objects. The application aims to allow interaction with image segmentation based on trained models.

KEYWORDS

BVLC / Caffe, CNN, clothing segmentation, conditional random fields, deep learning, image segmentation, neural networks

LUKAČOVIČ, Martin *Segmentace obrazu s využitím hlubokého učení*: diplomová práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 81 s. Vedúci práce bol Ing. Jan Mašek, Ph.D.

PREHLÁSENIE

Prehlasujem, že som svoju diplomovoú prácu na tému „Segmentace obrazu s využitím hlubokého učení“ vypracoval samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorském, o právoch súvisejúcich s právom autorským a o zmeně niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhé, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno ....

(podpis autora)

POĎAKOVANIE

Ďakujem vedúcemu mojej diplomovej práce Ing. Janovi Maškovi, Ph.D., za jeho čas venovaný konzultáciám a za cenné pripomienky a rady, ktoré mi pomohli pri vypracovávaní diplomovej práce. Taktiež by som sa rád poďakoval mojej rodine, ktorá mi bola oporou počas celého štúdia.

Brno ....

(podpis autora)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

POĎAKOVANIE

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno . . . . . .

. . . . . .

(podpis autora)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj
pro inovace

OBSAH

ZOZNAM OBRÁZKOV

ZOZNAM TABULIEK

ÚVOD

Táto práca sa zaoberá segmentáciou obrazu na úrovni pixelov a ich rozradením do tried pomocou prostredia pre hlboké učenie. So segmentáciou obrazu je možné sa stretnúť v dnešnej dobe napríklad v oblasti medicíny za účelom odhalenia možných chorôb pľúc či nádorov na mozgu na základe segmentácie medicínskych snímkov, ako pomôcka pre ľudí s čiastočnou stratou videnia či u autonómne riadených prostriedkov, kedy sa rozpoznáva segmentované prostredie.

Dôležitú úlohu z pohľadu princípu zohrávajú neurónové siete, preto je časť práce venovaná ich opisu. Existuje množstvo metód pre segmentáciu obrazu, v práci je zvolená metóda konvolučných neurónových sietí s ďalším spracovaním pomocou podmienených náhodných polí. Práca sa ďalej zameriava na preštudovanie a výber prostredia pre segmentáciu obrazu na úrovni pixelov, pričom na realizáciu segmentácie bolo využité prostredie pre hlboké učenie BVLC / Caffe.

V praktickej časti bolo najskôr potrebné nakonfigurovať systém spolu so súčasťami vyžadovanými pre prostredie hlbokého učenia. Pomocou vybraného prostredia bola potom overená funkčnosť voľne dostupného modelu pre segmentáciu obrazu. Ďalším výstupom je trénovanie na základe vybranej metódy a vytvorenie vlastných modelov. S cieľom lepšej interakcie užívateľa bolo navrhnuté webové rozhranie pre výber obrázku, vykonanie segmentácie a následného zobrazenia vysegmentovaného obrázku i s možnými triedami vzťahujúcich sa k rozpoznaným pixelom.

Hlavným prínosom práce je praktická realizácia vybranej metódy pre segmentáciu obrazu i s podrobne rozpísaným princípom a jej následnou implementáciou do webového rozhrania. Týmto spôsobom je docielené jednoduché a rýchle otestovanie natrénovaných modelov. Jedným z výsledkov práce je natrénovaný model na základe sady obrázkov VOC 2012 s možnosťou rozradenia do 20 tried a s dosiahnutou presnosťou 68,52 %. Pri trénovaní na základe sady obrázkov Fashion s možnosťou rozradenia do 18 tried dosahuje výsledný model presnosť 50,74 %.

Práca je členená následovne: prvá kapitola sa venuje súčasným metódam segmentácie obrazu s využitím hlbokého učenia ako i ostatným prístupom využívaných v praxi. Druhá kapitola popisuje vývoj neurónových sietí, vysvetlený je ich jednoduchý model, kapitolu uzatvára definícia konvolučných neurónových sietí. Tretia kapitola sa zaoberá princípom segmentácie obrazu pomocou využitia podmienených náhodných polí. V štvrtej kapitole je uvedený prehľad prostredí pre implementáciu hlbokého učenia a opis zvoleného prostredia pre realizáciu segmentácie. V kapitole sú uvedené niektoré parametre pre optimalizáciu trénovania. Piata kapitola sa zaoberá navrnutým webovým rozhraním. Šiesta kapitola opisuje parametre systému, štruktúru oboch datasetov a dosiahnuté výsledky, kedy sú vývoj a stratová funkcia zobrazené v grafoch. Siedma kapitola sa venuje diskusii dosiahnutých výsledkov.

1 SÚČASNÉ METÓDY SEGMENTÁCIE OBRAZU S VYUŽITÍM HLBOKÉHO UČENIA

Oblastí, v ktorých si neurónové siete s využitím hlbokého učenia nájdu svoje uplatnenie je veľmi veľa a zároveň sú veľmi rozmanité. V kapitole je zmienená popredná organizácia, ktorá sa venuje tejto problematike, základné rozdelenie počítačového videnia, ďalej budú uvedené používané typy neurónových sietí, metódy pri ktorých sa uplatňujú a konkrétne príklady využitia v praxi i s odkazom na publikáciu. Väčšia pozornosť je venovaná samotnej sémantickej segmentácii s využitím hlbokého učenia, nakoľko je témou tejto práce.

1.1 Súčasné prístupy a využitie v praxi

Počítačové videnie

Počítačové videnie [1] je možné definovať ako vednú disciplínu, ktorá sa snaží výpočtovými prostriedkami napodobniť ľudské videnie. Všeobecne je možné túto disciplínu rozdeliť do nasledujúcich kategórií abstrakcie:

- **nízka úroveň** – predspracovanie obrazu, nájdenie potrebných informácií pre vyššie úrovne, typicky jednoduché prvky obrazu ako okraje, prvky textúr ai.
- **stredná úroveň** – jednoduchý popis obrazu, napríklad hranice, objemy a plochy.
- **vysoká úroveň** – porozumenie obrazu, typicky klasifikácia objektu, scény alebo udalosti.

Organizácia ImageNet

ImageNet [2] je organizácia, ktorej snahou je poskytnúť všetkým výskumníkom vo svete databázu, ktorá bude kvalitná, obsahujúca dostatočné množstvo materiálu a ľahko dostupná. Počnúc rokom 2010, býva usporiadaná každoročná súťaž s názvom „Imagenet Large Scale Visual Recognition Challenge“ (ILSVRC), ktorej sa zúčastňujú jednotlivé výskumné tímy predkladajúc svoje programy a algoritmy pre klasifikáciu a detekciu objektov a scén. V roku 2011 sa pokladal za dobrý výsledok výstup s úrovňou chybovosti 25 %, v roku 2012 sa podarilo pomocou hlbokých konvolučných neurónových sietí dosiahnuť lepšieho výsledku a znížiť chybovosť na 16 %. Týmto spôsobom a dômyselnejšími metódami klesli úrovne chybovosti na niekoľko

[1] Z anglického významu Computer Vision
[2] http://image-net.org/

málo percent. Rozdelenie metód do kategórií, v ktorých sú porovnávané dosiahnuté výsledky je následovné [1]:

1. Lokalizácia objektu pre 1000 kategórií.
2. Detekcia objektu pre 200 plne značených kategórií.
3. Detekcia objektu z videa pre 30 plne značených kategórií.
4. Klasifikácia prostredia obrazu / scény pre 365 kategórii možných scén.
5. Analýza prostredia obrazu / scény pre 150 kategórií vecí a nesúvislých objektov.

Detekcia objektov

Príkladom systému pre detekciu objektov v obraze, sú tzv. rýchlejšie konvolučné neurónové siete (Faster R-CNN – Faster Region-based Convolutional Network). Tento systém kombinuje vlastnosti siete s oblastným navrhovaním (RPN – Region Proposal Network) a rýchle konvolučné neurónové siete pre detekciu. RPN sú taktiež plne konvolučné neurónové siete, ktoré súčasne predpovedajú hranice objektu a jeho stav na každej pozícii. Trénovaním RPN celou sieťou sú potom generované vysoko kvalitné oblastné návrhy, ktoré sú využité rýchlymi konvolučnými neurónovými sieťami. Systém vytvára dojem zjednotenej siete s účeľom detekcie objektov. Vzhľadom na rýchlosť a nízku nákladovosť je možné využiť tento systém v reálnom čase. Obšírnejšie vysvetlenie je možné nájsť v [2].

Sledovanie objektu

Vizuálne sledovanie tiež často označované sledovanie objektov, sa vzťahuje k automatickému odhadu trajektórie objektu, ktorý sa pohybuje vo videu. Uplatnenie má v mnohých oblastiach, vrátane kamerového dohľadu pre bezpečnosť, interakcie človeka s počítačom či u analýzy športového videa. Jednou z možností prevedenia takéhoto systému je podľa [3] online sledovací algoritmus učením diskriminačnej asymetrickej mapy s využitím konvolučných neurónových sietí. Tieto metódy sú potom použité pre učenie diskriminatívnych modelov na sledovanie objektu pomocou online systému podporných vektorov (SVM – Support Vector Machine).

Obrázok so super-rozlíšením

Ďalšou z oblastí využitia konvolučných neurónových sietí je tvorba obrázku s vysokým rozlíšením na výstupe, pričom na vstupe je obrázok s nízkym rozlíšením. V [4] je opísaný systém, kedy sa konvenčné metódy s riedkym kódovaním pre vysoké rozlíšenie obrázku úspešne pretransformolovali do oblasti hlbokých konvolučných neurónových sietí tzv. mapovaním. Aj napriek odľahčenému modelu CNN sa podarilo

dosiahnuť najvyššie výsledky z pohľadu kvality a náročnosti na výpočet, čo fakticky tento systém predurčuje k využitiu i online formou.

Detekcia hrán a okrajov

Publikácia [5] sa zaoberá princípom detekovania hrán a okrajov v obrázkoch zachytávajúcich prírodu. Táto problematika má taktiež dosah a tvorí základ v rôznych iných oblastiach ako je segmentácia, detekcia objektov, analýzy sledovania, zobrazovanie v oblasti medicíny, 3D rekonštrukcie, autonómne riadenie či prevod obrázku na text. V zmienenej publikácií je vysvetlený algoritmus, ktorý čerpá z vlastností plne konvolučných neurónových sietí a hlbokých sietí trénovaných pod dohľadom.

Sémantická segmentácia

Sémantická segmentácia je jednou z metód pre označovanie objektov a súčastí obrazu na princípe úrovne jednotlivých pixelov, resp. rozpoznaním pixelov. Využitím tejto metódy sa realizuje účeľové ťaženie z kapacity a vlastností techník hlbokého učenia. Samozrejme, vsetko má svoje limity a to i z pohľadu spomínanej kapacity hlbokého učenia vo vzťahu k opísaniu / vykresleniu objektov. Existuje veľké množstvo prístupov pre elimináciu týchto nedostatkov, všeobecne ich je možné kategorizovať to dvoch hlavných stratégií.

Prvá stretégia je založená na uplatnení oddelených mechanizmov pre extrakciu príznakov a segmentácie obrazu využívajúcej hrany a črty obrazu [6][7]. Praktickým prípadom je aplikácia CNN pre extrakciu významných častí obrazu a použitie superpixelov pre opis štruktúry obrazu. V [7] autori najskôr z obrazu získali superpixely, na ktorých následne použili proces extrakcie. Hlavnou nevýhodou tejto stratégie je, že chyby v prvých návrhoch (ako sú napr. spomínané superpixely) môžu viesť k zlým predikciám, bez ohľadu na to ako kvalitný je nasledujúci proces extrakcie.

Druhá stratégia je založená na princípe priameho učenia nelineárneho modelu z obrázkov do mapy tried. Príkladom je [8], kedy autori nahradili posledné plne pospájané vrstvy CNN konvolučnými vrstvami pre uchovanie priestorových informácií. Významným prínosom v tomto smere bola práca [9], v ktorej je opísaná aplikácia konceptu plne pospájaných konvolučných sietí, pričom najvyššie vrstvy získavali informácie o významných bodoch obrazu pre rozpoznanie objektov a najnižsie vrstvy informácie o štruktúre obrazu ako napr. hrany. Ich vzájomné prepojenie viedlo ku kombinácii týchto informácií. V publikáciách [10][11] boli použité podmienené náhodne polia (CRF – Conditional Random Field) pre spresnenie výsledkov získaných pomocou CNN s rôznym finálnym využitím. V oboch prípadoch boli CRF použité ako samostatný proces a teda oddelené od CNN, čím nedochádzalo ku trénovaniu celou sieťou. V kontraste s týmto prístupom bola implementovaná forma konvolučných

neurónových sietí, ktoré kombinujú CNN a CRF založené na pravdepodobnostnom grafickom modelovaní. Metóda pre dodatočné vyhľadenie výsledku pomocou podmienených náhodných polí je podrobnejšie opísaná v podkapitole 3.2 a metóda pre trénovanie kombinácie CNN a CRF ako zjednoteného CRF-RNN je podrobnejšie opísaná v kap. 3.

Vhodným príkladom pre porovnanie s vyššie uvedenými metódami je známy framework s názvom SegNet, ktorý je založený na odlišnom princípe. SegNet je architektúra, u ktorej sa uplatňuje využitie kódera a dekódera v oblasti sémantickej segmentácie s využitím hlbokého učenia pre viacpočetné triedy segmentácie na úrovni pixelov. Na ich výskume a vývoji sa podielali členovia skupiny Počítačového videnia a Robotiky [3] Cambridge Univerzity v Anglicku [12]. Ako bolo spomenuté, architektúra sa skladá zo sekvencie nelineárnych vrstiev (kóderov) a odpovedajúcej skupiny dekóderov, poslednou vrstvou je klasifikátor na úrovni pixelov. Typicky, každý enkóder obsahuje jednu či viac konvolučných vrstiev s tzv. „batch“ normalizátorom a nelineárnej jednotky ReLu, za ktorými nasleduje tzv. „maxpooling“ (združovacia vrstva) a podvzorkovanie. V časti dekóderu je riedke kódovanie, vznikajúce v dôsledku procesu združovania a nadvzorkovanie pomocou indexov zo združovacej vrstvy v časti kóderu. Toto nadvzorkovanie na základe indexov je vykonávané pre skvalitnenie rozlíšenia vytvorených máp, zachováva sa vysoká početnosť detailov v segmentovaných obrázkoch a redukuje sa celkový počet parametrov pre trénovanie v dekóderi. Trénovanie je možné celou architektúrou využitím metódy stochastickej aproximácie (niekedy označovanej aj ako inkrementálny alebo stochastický gradient descendant). Zámerom je získať z tohto modelu také predikcie, aby boli dostatočne hladké i bez použitia ďalšieho spracovania, napr. založeného na CRF [13].

Aplikovanie metód v praxi

Tzv. „Silent sound technology“ (SST) je technológia, ktorá spočíva v prenose informácií bez použitia hlasiviek. Princípom je transformácia pohybu pier do počítačom generovanej reči a následné šírenie sieťou. Vďaka tomu osoba na druhom konci hovoru dostáva informáciu v podobe generovaného zvuku. V tomto prípade sa uplatňujú vlastnosti hlbokých neurónových sietí [14].

Veľký pokrok bol zaznamenaný i v akustických modeloch, založených na hlbokých neurónových sietiach s uplatnením pri rozpoznávaní reči a príbuzných aplikáciach. V publikácii [15] sú zhrnuté hlavné výhody a prednosti neurónových sietí s využitím hlbokého učenia v porovnaní s predošlými riešeniami v tejto problematike.

[3] Z anglického významu Computer Vision & Robotics Group

Detekcia objektov a ich rozpoznanie sú najdôležitejšou témou v prevedení autonómne riadených prostriedkov. Hlavnou podstatou je detekovanie, sledovanie a následné rozpoznanie statických či dynamických objektov ako sú zvieratá, všetky druhy dopravných prostriedkov či chodcov. Problematika ma vysoký nárast v uplatňovaní v praxi a zaklad tvorí kombinácia hlbokých konvolučných neurónových sietí s odlišnými vrstvami a následným zavedením systému podporných vektorov [16].

Segmentácia nádoru na mozgu je dôležitou úlohou v spracovávaní medicínskych obrázkov. Skorá diagnóza nádoru je kľúčovou úlohou pri zvyšovaní možnosti vyliečenia, čím sa taktiež zvyšuje miera pravdepodobnosti prežitia u daného pacienta. Existuje veľa publikácií, ktoré sa venujú tejto problematike a ktoré zároveň využívajú odlišné metódy pre segementáciu na základe obrazov z magnetickej rezonancie. Akým spôsobom sa v tejto oblasti využívajú konvolučné neurónové siete a jednu z metód popisuje článok [17].

Ďalším príkladom sémantickej segmentácie s využitím hlbokého učenia v praxi je vývoj tzv. „Smart Specs“, čo je názov pre špeciálne okuliare navrhnuté pre ľudí, ktorí stratili zrak alebo pre čiastočne slepých ľudí, ktorí majú stále oblasti s pozostatkovým videním. Sú prispôsobené práve na tieto oblasti, resp. využitia ich maxima. Softvér pre ich funkciu je založený na princípe podmienených náhodných polí ako rekurentných neurónových sietí [18]. Okuliare dokážu vizuálne zobraziť prostredie a objekty, s ktorými prichádzame každodenne do styku, napr. stôl, stolička, kreslo atď. Excelujú i v tme, v rozpoznávaní veľkých objektov ako sú steny, značky a pod. [19]

Veľmi dobrým zdrojom pre rôzne druhy kategórii spracovávania obrazu je Model Zoo [4]. Tento zdroj obsahuje veľké množstvo odlišných metód spolu so sieťami i modelmi pre rôzne kategórie spracovávania obrazu ako je segmentácia, detekcia objetov, analýza scén ai.

[4] https://github.com/BVLC/caffe/wiki/Model-Zoo

2 HLBOKÉ UČENIE A NEURÓNOVÉ SIETE

Neurónové siete už od svojho vzniku fascinujú veľké množstvo vedcov, študentov či ľudí, ktorí radi experimentujú v oblasti schopností výpočtovej techniky a programovania. Pri bežnom postupe zostrojenia programu vývojár presne definuje počítaču čo robiť, rozkladá veľké problémy na menšie a určuje ďalšie úlohy, ktoré má daný program podniknúť pri riešení. Kontrastom takéhoto postupu sú neurónové siete, kedy neurčujeme počítaču každý krok a spôsob, ktorým má problém riešiť. Naopak, učí sa z pozorovaných dát a zisťuje pre seba najefektívnejšie riešenie v danej situácii. Neurónové siete sú základom implementácie, ktorá sa využíva v tejto práci. V kapitole bude popísaná história a vývoj neurónových sietí [20], zloženie, klasické modely a spôsoby učenia ako aj ich využitie v oblasti hlbokého učenia.

2.1 História a vývoj neurónových sietí

Rok 1943 je považovaný za vznik neurónových sietí ako takých, keď Warren McCulloch a Walter Pitts predstavili prvý model umelých neurónov, resp. matematický model neurónu. Parametre, ktoré bolo možné v tomto modeli nadobudnúť boli v rozmedzí $\{-1, 0, 1\}$. Týmto sa spustila vlna nových a ešte viac sofistikovaných riešení. Medzi jedných z bádateľov tej doby, ktorí sa nechali inšpirovať týmto modelom bol i zakľadateľ kybernetiky Norbert Wiener. Ďalším podstatním mylníkom bolo Hebbovo učiace pravidlo pre synapsiu neurónov – inými slovami medzineuronové rozhranie. Toto pravidlo bolo inšpirované myšlienkou, že pozorované podmienené rozhodovania u živočíchov sú už v skutočnosti vlastnosťou jednotlivých neurónov. Prvým zostrojeným neuropočítačom bol v roku 1951 „Snark“, za ktorého vybudovaním stál Marvin Minsky [20].

V roku 1957 Frank Rosenblatt vynašiel tzv. perceptron (viď podkapitola 2.3.1), ktorý zobecnil McCullochov a Pittsov model neurónu z pohľadu parametrov o množinu reálnych čísiel. Zároveň bol navrhnutý i učiaci algoritmus, o ktorom matematicky dokázal že po konečnom počte krokov nájde pre tréningové dáta odpovedajúci váhový vektor parametrov nezávisle na jeho počiatočnom nastavení. Rosenblatt spolu s Charlesom Wightmanom a ďalšími zostrojili neuropočítač, inšpirovaný architektúrou Snarku, navrhnutý pre rozpoznávanie znakov. Tento neuropočítač bol pomenovaný Mark l Perceptron. Vďaka úspešnej prezentácii neuropočítača sa dovtedy len alternatívne neurovýpočty začali implementovať a stali sa novým predmetom výskumu [20].

Po objave perceptronu bol vyvinutý ďalší typ neurónového výpočetného prvku, ktorý sa nazýval Adaline, skrátene z názvu „Adaptive Linear Element“. Jeho tvor-

com bol Bernard Widrow so svojou skupinou študentov a prvok bol obohatený o nové výkonné učiace pravidlo. Veľký úpadok v oblasti ďalšieho zdokonalovania neurónových sietí nastal po prehlásení Marvina Minskeho a Seymoura Paperta, ktorí v roku 1967 vyhlásili, že jeden perceptron nemôže počítať jednoduchú logickú funkciu, tzv. vylučovaciu disjunkciu (XOR), resp. že pre viacvrstvovú sieť s tromi neurónmi neexistuje a je nemožné vytvoriť učiaci algoritmus.

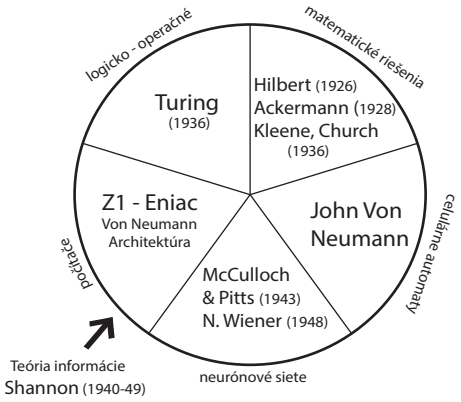
Po roku 1980 sa vďaka niektorým vedcom podarilo získať financovanie pre vývoj neurovýpočtov a neurónových sietí mnohými organizáciami. Publikovanie výsledkov skupiny bádateľov, tzv. „PDP“ skupiny (Parallel Distributed Processing Group) viedlo k vyriešeniu problému pomocou algoritmu spätného šírenia chyby „backpropagation“, pre viacvrstvovú neurónovú sieť. Neskôr sa ukázalo, že daný algoritmus bol publikovaný už v tichom období, kedy sa vývojom neurovýpočtov venovala len veľmi malá skupina vedcov. Praktický význam tohto algoritmu utvrdzuje systém NETtalk, vytvorený učením neurónovej siete z príkladov, ktorého činnosťou bolo konvertovanie anglického písaného textu do hovoreného.

Za zmienenie stojí i prvá väčšia konferencia v roku 1987 (IEEE International Conference on Neural Networks) špecializovaná na neurónové siete s účasťou 1700 ľudí [20].

2.2 Zloženie neurónových sietí

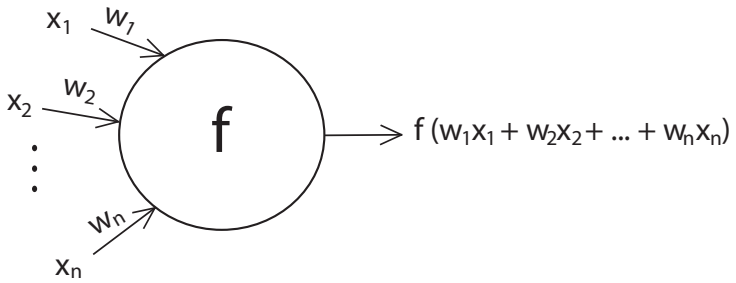
Neurónovú sieť možno definovať ako prepojenú zostavu jednoduchých procesných prvkov, jednotiek alebo uzlov, ktorých funkčnosť je voľne inšpirovaná na vlastnostiach neurónov zvierat či ľudí. Dá sa taktiež povedať, že skupina neurónov potom modeluje konkrétny problém približne rovnakým spôsobom ako skupina biologických neúronov spojených tzv. axónmi, teda rozvetvenými vláknami medzi neurónmi. Obecne, každý neurón je spojený s mnohými ďalšími a tieto spojenia môžu byť vyžiadané či naopak obmedzené podľa stavu každého z týchto pripojených neurónov. Tendenciou neurónových sietí je sklon k uchovávaniu experimentálnych znalostí a ich ďalšieho využitia.

Každá individuálna neurónová jednotka môže mať sčítaciu funkciu, ktorá kombinuje hodnoty všetkých vstupov hromadne. Možnosť výskytu prahovej alebo limitujúcej funkcie na každom spojení či na jednotke samotnej potom vyzýva k vyhodnoteniu danej situácie ešte pred ďalším šírením k ostatným neurónom. Preto platí tvrdenie, že neurónové siete sú skôr samoučné a trénovateľné ako explicitne programované a vynikajú v oblastiach, kde riešenie alebo presnú definíciu úlohy je obtiažne vyjadriť prostredníctvom bežných počítačových programov [21].



Obr. 2.1: Modely pre riešenie výpočetných problémov

Ako je možné vidieť na obrázku 2.1, neurónové siete je možné považovať za jednu z možností pri riešení výpočetných problémov. Modelov navrhnutých na riešenie úloh v tejto oblasti je celkom päť pričom neurónové siete spadajú pod biologický model. Tento model bol založený na princípe tých najdôležitejších aspektov fyziológie neurónov, čo vytvorilo základ pre modely umelej neurónovej siete, ktoré nepracujú sekvenčne ako je to napríklad v prípade Turingovho stroja. Neurónové siete majú hierarchickú viacvrstvovú štruktúru, ktorá ich odlišuje taktiež od celulárnych automatov, čo znamená že informácia je prenášaná nie len k najbližším susedným neurónom ale i na vzdialenejšie neurónové jednotky [21].



Obr. 2.2: Neurón ako primitívna funkcia

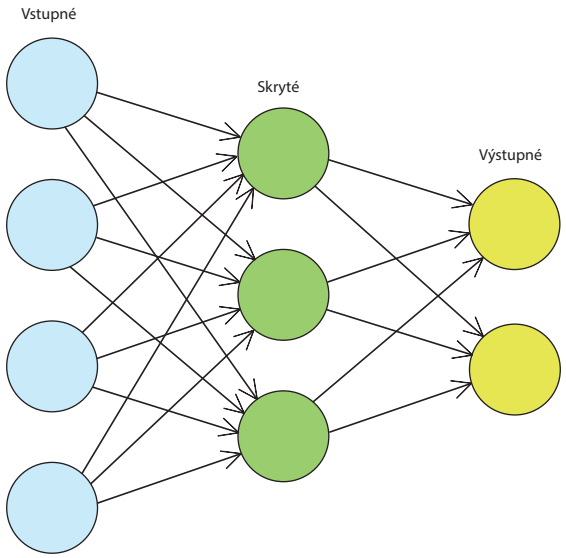
Na obrázku 2.2 je štruktúra neurónu s množstvom vstupov n. Každý vstupný kanál o dokáže preniesť skutočnú hodnotu x_o. Primitívnu funkciu f, ktorá sa využíva pre vykonanie operácie je možné zvoľiť ľubovoľne. Zvyčajne majú jednotlivé vstupné kanály priradenú tzv. váhu w, čo znamená že vstupná informácia x_o je násobená s odpovedajúcou váhou w_o [21].

Ak budeme uvažovať o každom uzle v umelej neurónovej sieti ako o primitívnej funkcii, ktorá je schopná transformovať svoj vstup na presne definovaný výstup, potom umelé neurónové siete možno označiť ako siete primitívnych funkcií. Rôzne modely umelých neurónových sietí sa líšia najmä v predpokladoch o využití daných funkcií, v spôsobe prepojenia a načasovania prenosu informácií [21].

Vo viacvrstvových štruktúrach neurónových sietí rozlišujeme jednotlivé vrstvy na:

- vstupnú vrstvu, v ktorej sú vstupy neurónov tvorené výľučne z vonkajšieho prostredia, výstup je obvykle smerovaný na ďalšie neuróny.
- skrytú vrstvu, v ktorej sú vstupy neurónov tvorené z výstupov ostatných neurónov alebo i z vonkajšieho prostredia cez prahové prepojenia, ich výstupy postupujú ďalej v neurónovej sieti.
- výstupnú vrstvu, je podobná skrytej vrstve avšak s tým rozdielom, že výstup z tejto vrstvy smeruje do vonkajšieho sveta [22].

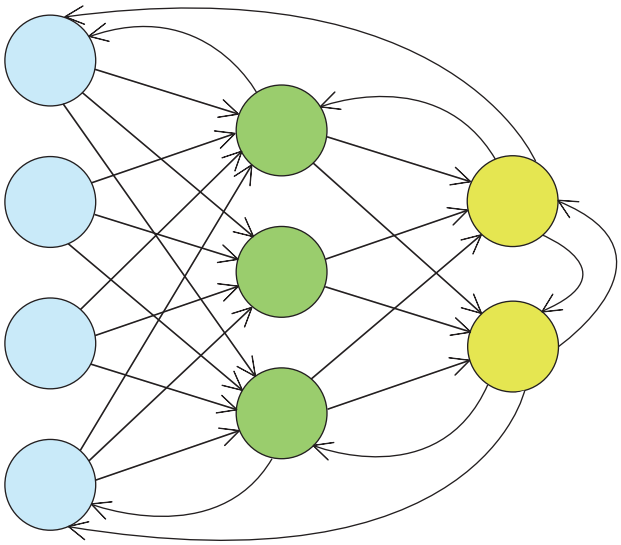
Na obrázku 2.3 je príklad viacvrstvovej neurónovej siete s topológiou doprednej neurónovej siete. Pri využití tohto typu topológie sa signál šíri po orientovaných synaptických pripojeniach len jedným smerom a to dopredu.



Obr. 2.3: Príklad viacvrstovej neurónovej siete s topológiou dopredných neurónových sietí

Na obrázku 2.4 je príklad viacvrstvovej neurónovej siete s topológiou rekurentnej neurónovej siete, pri ktorej sa vstupná a výstupná vrstva len veľmi ťažko odlišuje, nakoľko niekedy neuróny v rekurentných sietiach predstavujú vstupné ale aj výstupné

typy neurónov.



Obr. 2.4: Príklad viacvrstovej neurónovej siete s topológiou rekurentných neurónových sietí

2.3 Klasické modely neurónových sietí

Historicky prvý úspešný model neurónovej siete bola sieť perceptronov. V súčasnosti sa môžeme často stretnúť s využitím iných modelov umelých neurónov ako je napríklad sigmoid. Ich princíp a vzájomné odlišnosti sú popísané v nasledujúcich podkapitolách.

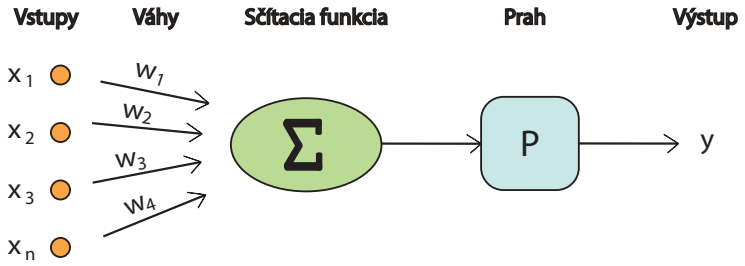
2.3.1 Perceptron

Ako už bolo zmienené v podkapitole 2.1, jedná sa o najjedodnoduchší model neurónu. Jednoduchý perceptron je na obrázku 2.5 a jeho prostým princípom je dichotomická klasifikácia, teda rozdelenie do dvoch tried, pri ktorých sa predpokladá, že sú lineárne separovateľné v príkladovom priestore [23].

Predstavme si niekoľko vstupov $x_1 ... x_n$, z ktorých každý má pridelenú odpovedajúcu váhu w – tá vyjadruje jeho dôležitosť. Hodnoty váh a prahu P sú vyjadrené reálnymi číslami, hodnota výstupu y nadobúda vždy hodnotu 0 alebo 1 a je určená po porovnaní výsledku sčítania $\sum_j w_j x_j$ s hodnotou prahu [23].

Matematické vyjadrenie výstupu takéhoto perceptronu je potom následovné:

$$y = \begin{cases} 0, & \sum_j w_j x_j > P \\ 1, & \sum_j w_j x_j \leq P \end{cases} . \tag{2.1}$$



Obr. 2.5: Perceptron

Pre komplexnejšie problémy sa využíva celá sieť perceptronov. Tentokrát si predstavme viacvrstvovú sieť, v ktorej prvá vrstva perceptronov vykonáva jednoduchšie rozhodnutia podľa zváženia vstupov. Pokračovaním do ďalšej vrstvy perceptronov sa dostávame ku komplexnejšiemu levelu rozhodovania – perceptrony teraz zvažujú výsledky z výstupov predchádzajúcich perceptronov. Týmto spôsobom je možné vykonávať viac sofistikované rozhodnutia [23].

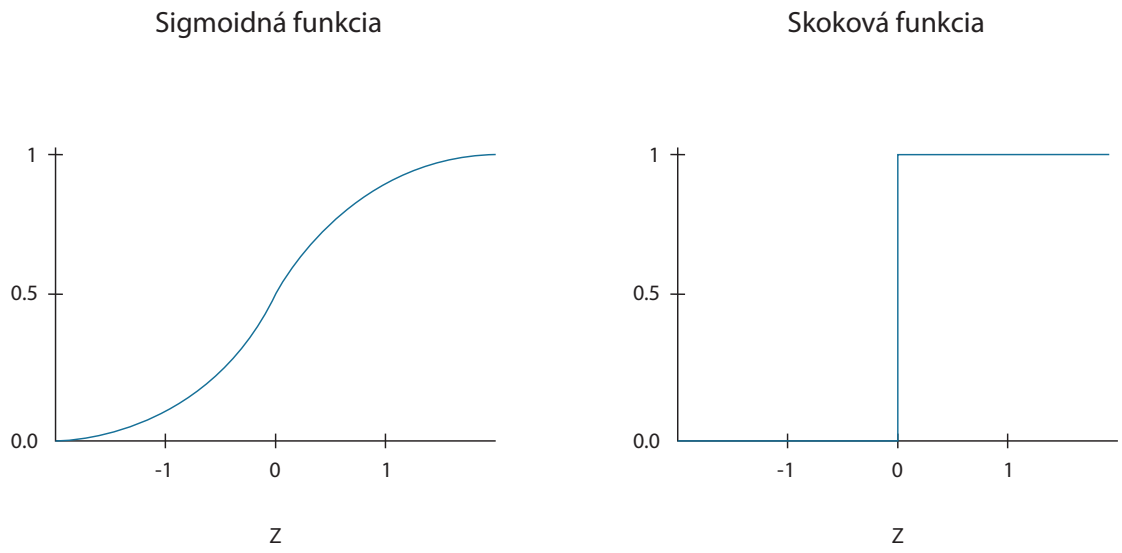
Zjednodušením výsledku sčítania $\sum_j w_j x_j$ na skalárny súčin $w \cdot x$ a pridaním biasu platí potom následovné matematické vyjadrenie:

$$y = \begin{cases} 0, & w \cdot x + b \leq 0 \\ 1, & w \cdot x + b > 0 \end{cases} . \tag{2.2}$$

Prah bol nahradený biasom a presunutý na druhú stranu nerovnosti. Bias možno definovať ako $b \equiv -P$. Bias sa privádza na vstup perceptronu, má vždy nenulovú, konštantnú hodnotu a zaručuje zmenu váh [23].

2.3.2 Sigmoid

Sigmoidné neuróny sú vhodným riešením v prípade ak chceme malou zmenou váh prispôsobiť správanie siete a teda celkový výstup na nami požadovanú hodnotu – v porovnaní so sieťou perceptronou, kde môže často nastať situácia kedy malou zmenou váh alebo biasu zapríčinime veľkú zmenu na výstupe perceptronu, napríklad z 0 na 1. Na obrázku 2.6 je možné vidieť rozdiel medzi sigmoidou ako prenosovou funkciou sigmoidných neurónov a skokovou funkciou v prípade perceptronu.



Obr. 2.6: Sigmoidná funkcia a skoková funkcia

Princíp sigmoidu je potom následovný: opäť máme niekoľko vstupov $x_1, \ldots, x_n$, z ktorých každý má pridelenú odpovedajúcu váhu w. Vstupy však nadobúdajú hodnoty nielen 0 alebo 1 ale zväčša hodnoty medzi nimi ako napríklad 0,449. Výstupom y je potom $\sigma(w \cdot x + b)$ kde σ je sigmoidná funkcia vyjadrená:

$$\sigma(y) \equiv \frac{1}{1 + e^{-y}}. \tag{2.3}$$

Pri rozhodovaní je potom možné stanoviť hodnotu napr. vyššiu ako 0,5 za vyhovujúcu naším potrebám a naopak hodnotu menšiu ako 0,5 za nevyhovujúcu. Obecne je možné sigmoidnú funkciu nahradiť za inú funkciu pri zachovaní princípu, medzi ďalšie používané funkcie patrí napríklad hyperbolická tangenta, lineárna, Gaussová ai [23].

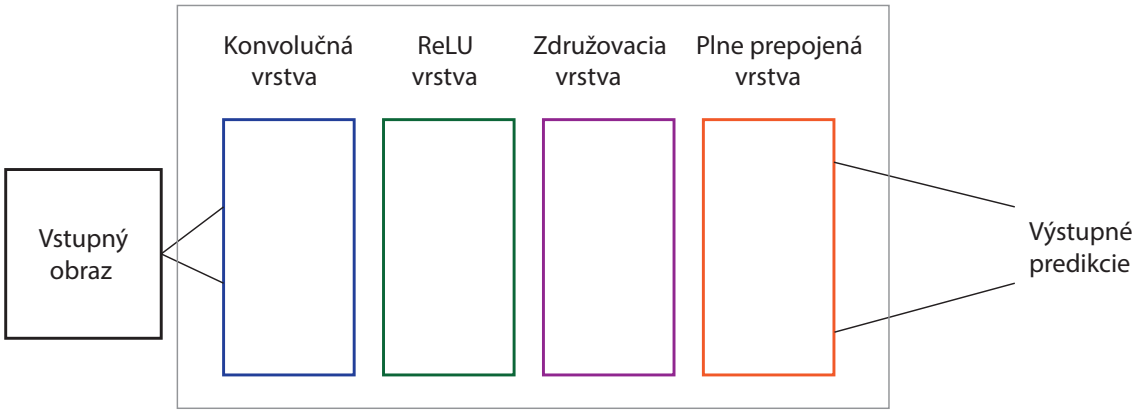
2.4 Hlboké učenie

V počiatočných dňoch umelej inteligencie sa postupne veľmi rýchlo podarili vyriešiť problémy, ktoré sú pre človeka intelektuálne náročné ale pomerne jasné pre počítače – problémy, ktoré sú opísané skupinou formálnych matematických zákonov. V porovnaní s tým, je náročné opísať problémy, ktoré človek rieši intiutívne, automaticky ako je napríklad rozpoznanie hovoreného slova či rozpoznanie tvárí v obraze a tieto úlohy vyriešiť pomocou počítača. Východiskom je v tomto prípade istý druh učenia umelej inteligencie zo skúsenosti a porozumenia konkrétneho problému.

Hlboké učenie je forma strojové učenia, založená na koncepte neurónových sietí. Najvýznamnejšie a najvyužívanejšie siete v tejto oblasti sú konvolučné neurónové siete. Ako už bolo v predošlej časti textu zmienené, siete môžu mať vstupné, skryté a výstupné vrstvy. Výraz hĺbka sa označuje za technický termín, ktorý vyjadruje využitie viac ako jednej skrytej vrstvy. V prípade, že sieť obsahuje viac ako tri vrstvy (vrátane vstupnej a výstupnej) kvalifikuje sa pojem hlboké učenie. V sietiach s hlbokým učením je každa vrstva zložená z jednotlivých uzlov (neurónov) a trénovaná na odlišnej sade funkcií, založených na výstupe predchádzajúcej vrstvy. Čím hlbšie sa sieťou prechádza tým komplexnejší rozsah dokážu jednotlivé uzly rozoznať vzhľadom na to, že sa naučené funkcie a zručnosti uzlov spájajú a kombinujú zo všetkých predošlých vrstiev [23].

2.5 Konvolučné neurónové siete

U konvolučných neurónových sietí (CNN – Convolutional Neural Networks) si neuróny hierarchicky vymieňajú spracovávanú informáciu. Konvolučné neurónové siete boli spočiatku navrhnuté a využité pre rozpoznávanie písaného písma rukou [24]. Vďaka ich vlastnostiam sa však využívajú i v dnešnej dobe pri riešení úloh na dvojrozmerných obrazových dátach ako je klasifikácia či triedenie [24]. Počas trénovania sa vytvorí reprezentácia obrázka, ktorého komplexnosť sa postupne stupňuje od rozpoznaniach svetlých či tmavých miest až po samotný význam a hodnotenie situácie. Konvolučné neurónové siete rozoberajú obraz ako maticu pixelov [24].



Obr. 2.7: Architektúra jednoduchej konvolučnej neurónovej siete

Všeobecne je architektúra konvolučných neurónových sietí definovaná na obr. 2.7. Štruktúru tvorí viac vrstiev – po vstupnej vrstve nasleduje konvolučná vrstva, z ktorej názvu vyplýva, že využíva matematickú metódu konvolúcie. Obraz sa systema-

ticky prechádza aplikovanými filtrami (označované tiež ako kernely) a využíva malé okolie na výpočet novej hodnoty. Za konvolučnou vrstvou nasleduje vrstva ReLu ako aktivačná funkcia, napríklad funkcia normalizácie $max(0, x)$, dvojíc vrstiev tohto druhu môže byť niekoľko. Následujúcou vrstvou je združovacia vrstva, v ktorej je vykonávane podvzorkovanie. Zastúpenia obrazu sú tak žmenšené a sústredené na významné časti. V určitom momente, po dosiahnutí kvalitatívneho zlúčenia do malého obrazu, prechádza výstup do plne prepojených vrstiev. V súčasnosti sú CNN využívané pri rozpoznávaní tvárí, objektov či dopravných značiek u autonómne riadených dopravných prostriedkov [25].

Ďalšie varianty hlbokého učenia pri využití neurónových sietí:

- Hlboké konvolučné neurónové siete – tieto siete sa líšia od základných konvolučných neurónových sietí vo väčšom počte skrytých vrstiev, čo znamená aj väčšiu náročnosť, v súčasnosti sa však využívajú najviac nakoľko napreduje i sila výpočtovej techniky [26].
- „Deep belief“ konvolučné neurónové siete – topológia siete a princíp má základ u konvolučných neurónových sietí, pracujú so štruktúrou 2D obrázkov a navyše sa tu využíva pred-trénovanie v deep belief sieti [24].
- Reziduálne siete – framework určený pre trénovanie veľmi hlbokých sietí. Reziduálne siete sú témou publikácie [27], zároveň sa stali víťazmi súťaže ILSVRC 2015. V porovnaní s architektúrou konvolučných neurónových sietí, zakončenie topológie netvoria plne prepojené konvolučné neurónové siete. Podľa [27] vykazujú nízku chybovosť pri aplikácii veľmi hlbokej siete (152 vrstiev). Metóda s využitím reziduálnych sietí má uplatnenie v oblasti klasifikácie obrazu, detekovania ojektu v obraze, sémantickej segmentácii a pod.

3 SEGMENTÁCIA OBRAZU S VYUŽITÍM HLBOKÉHO UČENIA

Ako už bolo spomínané v časti práce o aplikácii súčasných metód 1.1, sémantická segmentácia s využitím techník hlbokého učenia našla svoje uplatnenie v oblasti autonómne riadených vozidiel, možnosti rozpoznania objektov pre čiastočne slepých ľudí či ako pomoc v medicíne pri určovaní diagnóz zo snímkov. Existujú rôzne metódy ako využiť segmentáciu obrazu pri pixelovo orientovaných úlohách, základom sú však hlboké konvolučné neurónové siete.

Využitie hlbokých konvolučných neurónových sietí malo úspech v porozumení počítačom spracovávaných úloh v oblasti digitálnych obrázkov alebo videa, vhodným príkladom je rozpoznávanie obrazu [26] či detekcia objektov [28]. Tento úspech sa stal motiváciou pre využitie CNN v problematike pixelovo orientovaných úloh. Ďalšou kľúčovou podstatou je učenie celou sieťou „end-to-end“. Prispôsobenie takýchto CNN, ktoré vynikali v oblasti detekcie objektov v obraze má však niekoľko výziev, ktoré je nutné prekonať a uspôsobiť pre využitie na úrovni pixelov.

3.1 Sémantická segmentácia na princípe CRF-RNN

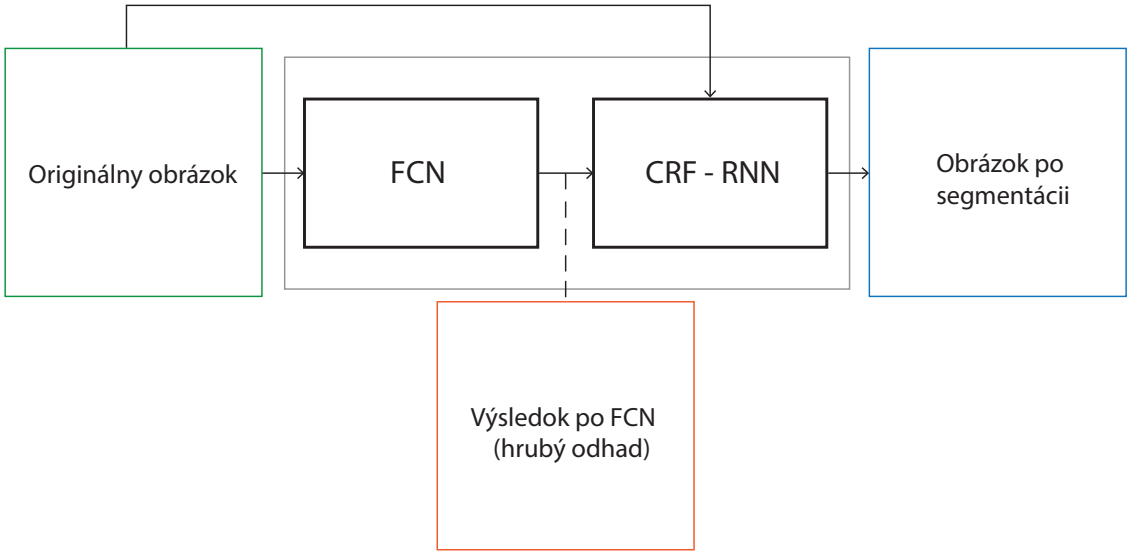
Predovšetkým platí fakt, že u bežných CNN sa využívajú konvolučné filtre s veľkými recepčnými poliami – to má za následok produkciu len veľmi hrubých výstupov obrazu pred samotným spracovaním na úrovni pixelov. Ďalšou súčasťou je prítomnosť podvzorkovacích vrstiev, čím sa ešte viac redukuje šanca pre kvalitný výstup segmentácie. Tieto a iné vlastnosti môžu spôsobiť neostré hranice a rozpité tvary pri úlohách týkajúcich sa sémantickej segmentácie. V druhom rade CNN chýbajú mechanizmy pre lepší spôsob odlíšenia dvoch veľmi podobných pixelov, resp. ich priradenie do konkrétnej triedy. Výsledkom sú potom nesprávne rozpoznané hranice objektov a súčastí obrazu či rôzne rušivé úseky vo výstupnom obraze po segementácii [18].

Otázkou teda je, ako využiť CNN v oblasti predikcie štrukturovaných výstupov. Všeobecne existuje viac spôsobov ako zdokonaliť vykreslovanie obrazu pomocou CNN pri pixelovo orientovanom prístupe, pričom opisovaná metóda CRF-RNN je založená na princípe vyznačenom v následujúcej časti textu hrubým písmom [29]:

1. Využitie druhu konvolúcie
2. ParseNet
3. Nezdružené (dekonvolučné) siete („Unpooling network“)
4. Dodatočné spracovanie pomocou podmienených náhodných polí
5. **Trénovanie celou sieťou s využitím podmienených náhodných polí**

Prítomnosť združovacej vrstvy v CNN, opísanej v podkapitole 2.5, ešte viac redukuje šancu na získanie výstupu po segmentácii vo vhodnej kvalite.

K odstráneniu spomínaných limitujúcich faktorov CNN, sa v niektorých metódach využívajú pravdepodobnostné grafické modely. Tie zohrávali dôležitú úlohu v pokroku techník hlbokého učenia a boli zároveň vyvinuté ako účinné prostriedky pre správnosť označenia jednotlivých súčastí obrazu na úrovni pixelov. Do popredia sa dostali najmä Markove náhodné polia (MRF – Markov Random Fields) a ich varianta – podmienené náhodné polia (CRF – Conditional Random field), ktoré zaznamenali najväčší úspech v tejto oblasti a stali sa najúspešnejšími a najčastejšie využívanými grafickými modelmi. Dá sa povedať, že svojimi vlastnosťami majú schopnosť rozpoznať, resp. objasniť súvislosti v obraze.



Obr. 3.1: Bloková schéma siete pre sémantickú segmentáciu obrazu metódou CRF-RNN

3.1.1 Podmienené náhodné polia CRF

Hlavnou myšlienkou využitia CRF pre sémantické označovanie je formulovanie problematiky spracovania obrazu na úrovni pixelov ako problematiky pravdepodobnostnej dedukcie, ktorá zahŕňa predpoklady ako je napr. správne rozhodnutie o priradení podobných pixelov do jednotlivých tried. Takáto dedukcia modelom CRF smeruje k upresneniu nevýraznych či naopak hrubých predikcií tried na úrovni pixelov čím je dosiahnutá produkcia ostrých a presnejšie vymedzených hrán a jemnozrnných segmentácií. Týmito vlastnosťami je možné eliminovať nevýhody využitia CNN pre úlohy označovania na úrovni pixelov [18].

Aplikácia CRF sa dá taktiež realizovať rôznymi spôsobmi. Jedným zo spôsobov pre zlepšenie výsledkov sémantického označovania na úrovni pixelov je aplikácia CRF až v ďalšom kroku, teda až po natrénovaní CNN, dôsledkom čoho sú úplne oddelené od tréningu CNN. Tento spôsob využíva metóda Deeplab, ktorej princíp je priblížený v podkapitole 3.2. Pre efektívnejšie využitie sily a účinnosti, ktoré CRF poskytujú, sa v metóde CRF-RNN uplatňuje ich integrácia v plnej hĺbke siete. Sieť je založená na kombinácii CNN spolu s CRF a trénovanie je realizované celou sieťou pre dosiahnutie lepších výsledkov – spájajú sa silné stránky CNN a CRF grafických modelov do zjednoteného rámca [18].

Matematické vyjadrenie

Pre priblíženie princípu sú pomenované jednotlivé premenné následovne [29]: X_i sa vzťahuje k pixelu i a vyjadruje triedu priradenú konkrétnemu pixelu i. Nadobúdať môže hodnoty z pred-definovaných hodnôt tried $L = \{l_1, l_2, ..., l_L\}$. Triedou môže byť napríklad pozadie, mačka, strom, osoba a pod. Nech X je vektor, tvorený náhodnými veličinami $X_1, X_2, ..., X_N$, kde N vyjadruje počet pixelov v obrázku. Graf $G = (V, E)$, kde $V = \{X_1, X_2, ..., X_N\}$ a globálne pozorovanie (obrázok) I, pričom dvojica (I, X) môže byť modelovaná ako CRF pomocou Gibbsovej distribúcie formulovaním následovného $P(\mathrm{X} = \mathrm{x}|\mathrm{I}) = \frac{1}{Z(I)}\exp(\text{-}\mathrm{E}(\mathrm{x}|\mathrm{I}))$. Tu sa $E(x)$ nazýva energia konfigurácie $\mathrm{x} \in L^N$ a $Z(\mathrm{I})$ je podielová funkcia. Maximalizácia pravdepodobnosti správneho priradenie pixelov je riešená minimalizáciou energie $E(x)$. V úplne pospájaných párových modeloch je energie pre priradenie tried x vyjadrená ako $E(x)$ = energia jednočlennej zložky + energia párových zložiek [30]. Matematické vyjadrenie:

$$E(x) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j). \tag{3.1}$$

Energia jednočlenných (unárnych) zložiek $\psi_u(x_i)$ vyjadruje opak pravdepodobnosti, a teda náklady (cenu) spojené s priradením triedy pre každý pixel nezávisle na ostatných. Energia párových zložiek vyjadruje náklady potrebné na priradenie triedy x_i a x_j, pixelom i, j súčasne. Jednočlenné zložky sú v našom prípade získavané z CNN, ktoré vyjadrujú predpovede rozradenia pixelov do tried a sú vykonané bez ohľadu na podobnosť či konzistencie tried. Energie párových zložiek poskytujú istý spôsob vyhladenia, závislý na dátach z obrazu, čo prispieva k priradeniu tried pixelom s obdobnými vlasnosťami [30].

3.1.2 Iterácia metódou stredného poľa

Individuálne kroky algoritmu iterácie sú rozložené a opísané ako vrstvy CNN. Iterácia je založená na využití filtrov, konkrétne Gaussovho priestorového a dvojstranného

filtru. Jednou z výhod ich využitia je nenáročnosť na parametre i napriek tomu, že môžu nadobúdať rozmery ako samotný obraz. V nasledujúcom texte sú formulované kroky iterácie ako CNN podľa [18].

Inicializácia

Počnúc touto definíciou, sa ďalej bude $U_i(l)$ používať v texte na vyjadrenie negatívnej unárnej energie $U_i(l) = -\psi_u(X_i = l)$. Vykonanie inicializácie je ekvivalentom k aplikácii softmax funkcie na všetkých unárnych potenciáloch naprieč všetkými triedami pre každý pixel.

$$Q_i(l) \leftarrow \frac{1}{Z_i} exp\left(U_i(l)\right) \ldots \text{pre všetky i}, \;\; Z_i = \sum_l exp\left(U_i(l)\right). \tag{3.2}$$

Šírenie správy

Vykonáva sa aplikovaním Gaussových filtrov M na hodnotách marginálnej distribúcie Q. Koeficienty Gaussovho filtru sú odvodené z prvkov obrazu ako napr. umiestnenie pixelov, RGB hodnoty ai. To odzrkadluje súvis medzi jednotlivými pixelmi. Počas spätného šírenia sú chyby odvodené na základe výstupov použitých Gaussových filtrov v opačnom smere.

$$\widetilde{Q}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k^{(m)}(f_i, f_j) Q_j(l) \ldots \text{pre všetky m.} \tag{3.3}$$

Štatistické váženie

Vychádza sa z váhového súčtu na výstupe filtrov M z predchádzajúceho kroku iterácie a to pre každú triedu l. V prípade, že sa ku každej triede pristupuje individuálne, vychádza sa z bežnej konvolúcie s použitím filtra 1x1, s počtom M vstupných kanálov a jedným výstupným. Vzhľadom na to, že vstupy i výstupy k tomuto kroku sú známe počas algoritmu spätného šírenia, je možné spočítať deriváciu chyby váh a tým umožniť automatické učenie váh filtrov.

$$\check{Q}_i(l) \leftarrow \sum_m w^{(m)} \widetilde{Q}_i^{(m)}(l). \tag{3.4}$$

Zosúladenie

Výstupy z predchádzajúceho kroku (označený v našom prípade $\check{Q}$) sú zdieľané medzi triedami vždy v určitej miere v závislosti na ich kompatibilite, tá je parametrizovaná funkciou $\mu(l, l')$. V tomto prípade sa využíva tzv. „Pottsov“ model, daný $\mu(l, l') = [l \neq l']$, kde hodnota vnútri $[.]$ udáva hodnotu „penalizácie“ ak sa do odlišných tried označia pixely s podobnými vlastnosťami. Dôležitou vlastnosťou je schopnosť určiť

výšku takejto penalizácie, napríklad pri priradení tried človek a bicykel podobným pixelom (s menšou penalizáciou) a tried človek a oblak (s väčšou penalizáciou).

$$\hat{Q}_i(l) \leftarrow \sum_{l' \in L} \mu(l, l') \check{Q}_i(l'). \tag{3.5}$$

Pridanie jednozložkového potenciálu

Výstupu z kroku zosúladenia je odčítaný prvok po prvku z unárneho vstupu U. Unárny vstup je poskytovaný v našom prípade z CNN. Pokým nie sú zahrnuté žiadne parametre, odvodené chyby spätným šírením sú jednoducho prevzaté z výstupu na oba vstupy, s príslušným označením.

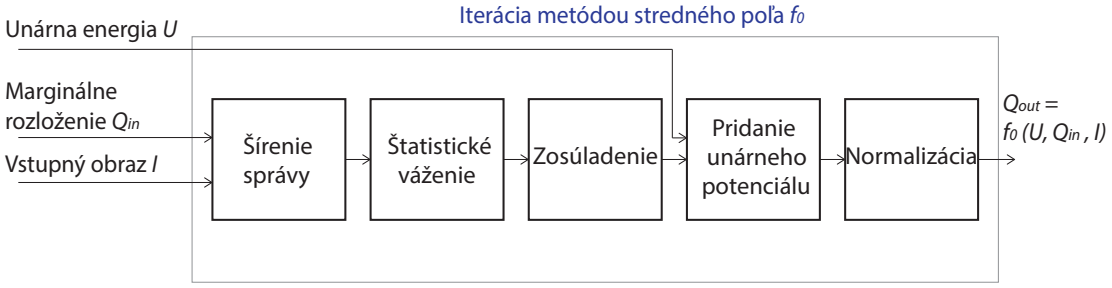
$$\check{Q}_i(l) \leftarrow U_i(l) - \hat{Q}_i(l). \tag{3.6}$$

Normalizácia

Posledný krok iterácie sa dá považovať ako ďalšie využitie softmax funkcie bez žiadnych udaných parametrov. Šírenie zisteného diferenciálu je zabezpečené softmax operáciou spätného šírenia.

$$Q_i \leftarrow \frac{1}{Z_i} exp\,(\check{Q}_i(l)). \tag{3.7}$$

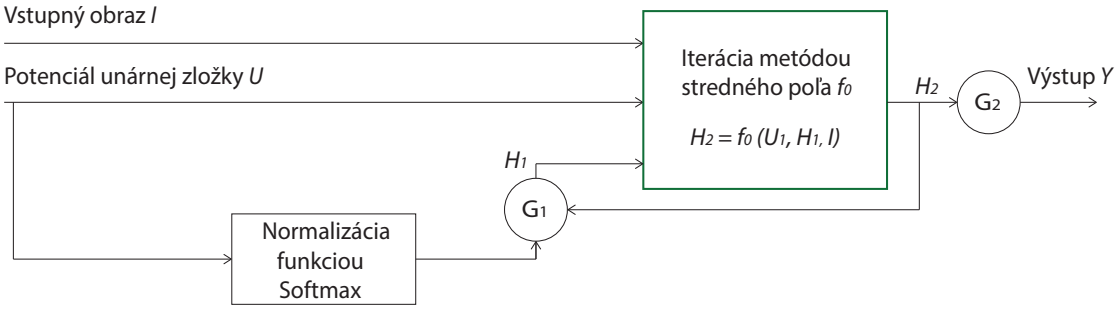
Jednotlivé časti iterácie je možné vidieť na obr. 3.2. Funkcia f_θ vyjadruje výsledok po priebehu jednej iterácie metódou stredného poľa, vektor $\theta = \{w^{(m)}, \mu(l, l')\}$, $m \in \{1, \ldots, M\}$, $l, l' \in \{l_1, \ldots, l_L\}$ zastupuje parametre iterácie.



Obr. 3.2: Jedna iterácia podľa metódy stredného poľa využívaná v CRF, môže byť modelovaná ako naskladané vrstvy CNN siete

3.1.3 Trénovanie siete

Ako už bolo v predošlom texte spomenuté, minimalizácia CRF energie $E(x)$ poskytuje najpravdepodobnejšie priradenie triedy konkrétnemu pixelu x pre daný spracovávaný obrázok. Avšak vzhľadom na ťažkú riešiteľnosť tejto minimalizácie, CRF distribúcia $P(\mathrm{X})$ je aproximovaná jednoduchou distribúciou $Q(\mathrm{X})$, čo je možné zapísať ako výsledný produkt nezávislých marginálnych distribúcií $Q(\mathrm{X}) = \prod_i Q_i(X_i)$ [29].



Obr. 3.3: Princíp fungovania algoritmu iterácie podľa metódy stredného poľa v podobe RNN

Mnohonásobné iterácie metódou stredného poľa sú implementované opakovaním vrstiev iterácií popísaných v predošlej podkapitole tým spôsobom, že každá iterácia nadobúda hodnoty Q z predchádzajúcej iterácie a unárne hodnoty v ich pôvodnej forme. V tomto prípade možno hovoriť o využití iterácií stredného poľa ako rekurentné neurónové siete (RNN – Recurrent Neural Networks). Akonáhle sa spustia výpočty v časti CRF-RNN, dáta sú spracovávané v počte T iterácií, teda vytvorenej smyčke prostredníctvom RNN. V opačnom smere, počas spätného šírenia, akonáhle sú dostupné chyby na výstupe Y, podobne potrvá T iterácií v rámci smyčky pred dosiahnutím RNN vstupu U s cieľom propagácie chýb ku CNN, ktoré poskytujú unárny vstup. Správanie zapojenia z obr. 3.3 je možné popísať nasledujúcimi rovnicami [18] [30]:

$$H_1(t) = \begin{cases} softmax(U), & t = 0 \\ H_2(t-1), & 0 < t \leq T, \end{cases} \tag{3.8}$$

$$H_2(t) = \begin{cases} f_\theta(U, H_1(t), I), & 0 \leq t \leq T, \end{cases} \tag{3.9}$$

$$Y(t) = \begin{cases} 0, & 0 \leq t \leq T \\ H_2(t), & t = T, \end{cases} \tag{3.10}$$

kde T je počet iterácií podľa metódy stredného poľa.

Zhrnutím doteraz uvedených faktov v kapitole platí následovné: metóda CRF-RNN je formulovaná aproximáciou metódy stredného poľa pre husté CRF spolu s určením Gaussového párového potenciálu ako RNN, ktoré dokážu zlepšiť kvalitu hrubého výstupu CNN v poprednom šírení, zatiaľ čo sa chybový diferenciál vracia späť počas trénovania CNN. Metóda je založená na možnosti trénovania siete v celej hĺbke, čo zahŕňa CNN i RNN pre dedukciu z CRF, využitím známeho algoritmu spätného šírenia. Týmto spôsobom sa využívajú vlasnosti ako hlbokého učenia tak i grafického modelovania.

3.2 Sémantická segmentácia na princípe Deeplab

Hlboké konvolučné neurónové siete (DCNN – Deep Convolutional Neural Networks) sa uplatnili pri riešení náročných počítačom-spracovávaných grafických úlohách a boli dôležitou súčasťou pri dosiahnutí najlepších výsledkov za posledné obdobie vývoja v tejto oblasti.

Pri využití DCNN na úlohy spojené so sémantickou segmentáciou je najskôr potrebné odstrániť určité prekážky pre dosiahnutie očakávaných výsledkov. Medzi dve najväčšie z nich patrí:

1. Straty spôsobené podvzorkovaním
2. Problematika priestorovej intenzity

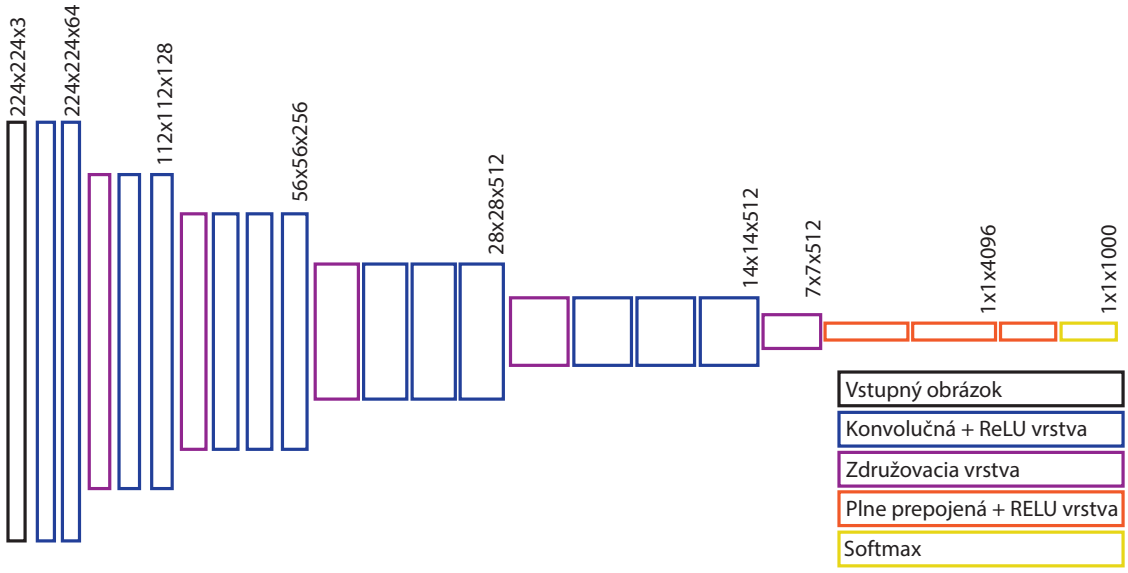
V prvom prípade sú straty spôsobené opakovanou kombináciou združovania a podvzorkovania v každej vrstve štandardných DCNN. Pre odstránenie tohto problému sa v metóde Deeplab používa zvláštny druh konvolúcie tzv. „atrous konvolúcia“, taktiež často označovaná ako algoritmus dier. Tento algoritmus bol pôvodne navrhnutý pre počítanie diskrétnej vlnovej transformácie bez podvzorkovania [31]. Zmienené riešenie zefektívňuje výpočty s hustým rozdelením z výstupov každej vrstvy DCNN v systéme [11].

Priestorová intenzita sa v DCNN vyznačuje vstavanou stálosťou (nemennosťou) voči lokálnej transformácii obrazu a je základom pre schopnosť učenia sa hierarchickej abstrakcie dát. Svoje opodstatnenie má najmä pri spracovávaní na úrovni vysokej náročnosti úloh ako je napr. jemnozrnná kategorizácia či klasifikácia objektov. Táto vlastnosť však môže predstavovať prekážku z pohľadu spracovávania úloh na nižšej úrovni ako je napr. odhad pozície objektu či z pohľadu samotnej sémantickej segmentácie, kedy sa uprednostňuje možnosť presnej lokalizácie ako abstrakcie priestorových detailov. V tom prípade hovoríme o určitej limitácii z pohľadu priestorovej presnosti, čo sa v rámci DCNN následne prejaví v poslednej vrstve vo forme spomínanej nedostatočnej lokalizácie pre presnosť segmentácie objektu. Elimináciu takéhoto problému je možné docieliť kombináciou DCNN a grafických pravdepo-

dobnostných modelov, kedy sú v konečnom výstupe zachytené i jemnejšie detaily a zároveň presnejšie rozpoznanie pixelov do tried [11].

3.2.1 Spôsob spracovania

Základným inicializačným modelom, na ktorom je metóda Deeplab založená je VGG-16 – verejne dostupná verzia šesťnásťvrstvového modelu pre klasifikáciu. Sieť tohto modelu je na obr. 3.4 spolu s hodnotami vo formáte *rozmery obrázka* x *počet filtrov*. Postupnou úpravou modelu tzv. „ladením“ vznikne efektívny systém pre sémantickú segementáciu obrazu s hustým rozdelením. Správne vyhodnotenie priestorových bodov je kľúčovým faktorom pre extrakciu príznakov v CNN s hustým rozdelením [11] [32].

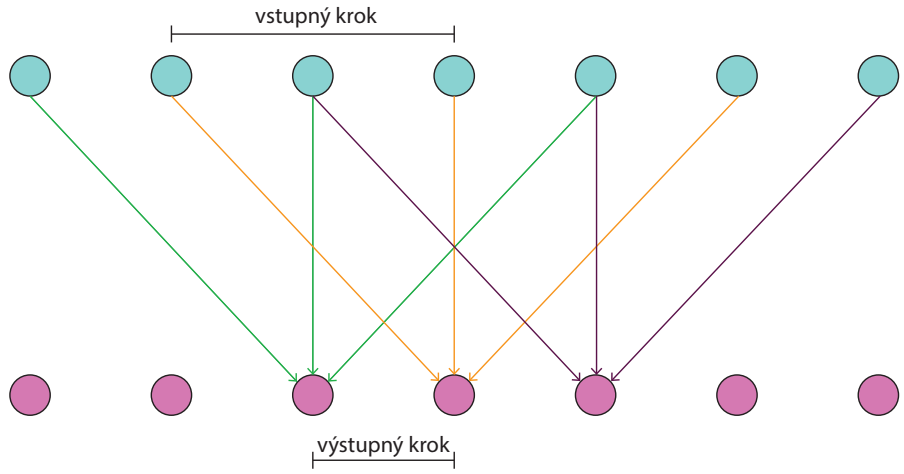


Obr. 3.4: Konvolučná neurónová sieť podľa VGG-16 pre klasifikáciu, ktorá sa upravuje pre možnosť sémantickej segmentácie

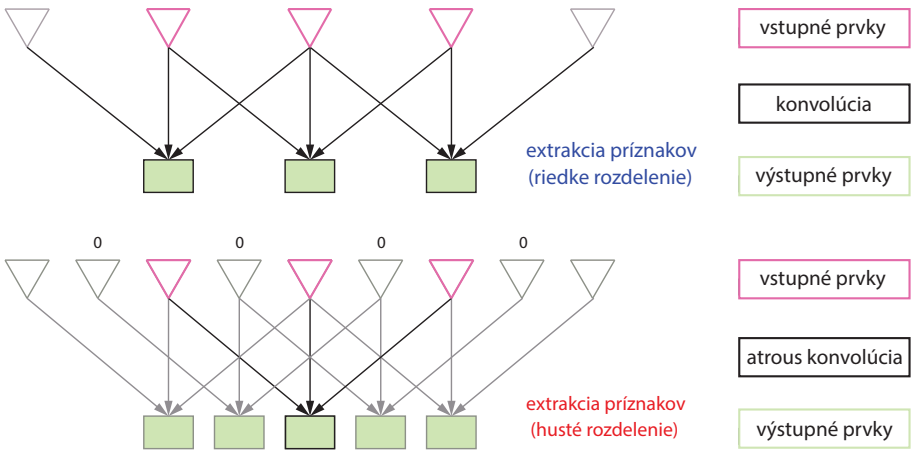
Jednou zo základných zmien siete je prepracovanie plne spojených vrstiev vo VGG-16 na konvolučné vrstvy. Efekt po tomto kroku však stále nie je postačujúci, nakoľko prináša vypočítané detekčné vyhodnotenia - skóre s riedkym rozdelením (s krokom tridsaťdva pixelov). Jednou z variánt pre výpočty s hustým rozdelením tohto skóre a s cieľovým krokom osem pixelov je možné realizovať vynechaním nadvzorkovania po posledných dvoch združovacích vrstvách siete a úpravou konvolučných filtrov v následujúcich vrstvách zavedením núl pre zväčšenie ich dĺžky. Efektívnejším spôsobom je ponechanie filtrov bez zmeny a zavedenie riedkeho výberu z mapy, na ktorej boli aplikované so vstupným krokom dva pixely, respektíve štyri pixely [11].

Algoritmus dier

Princíp tohto druhu konvolúcie je zobrazený na obr. 3.5. V prípade obr. 3.6 je možné vidieť rozdiel v uplatnení štandardnej konvolúcie na výbere (extrakcie) príznakov s riedkym rozdelením, kedy vstup tvorí mapa s nízkym rozlíšením. Nižšie v tomto obrázku je zobrazený algoritmus dier v prípade extrakcie príznakov s hustým rozdelením, kedy vstup tvorí mapa s vysokým rozlíšením. Platí, že veľkosť filtru je 3, veľkosť vstupného kroku je 2 a veľkosť výstupného kroku je 1 [11].



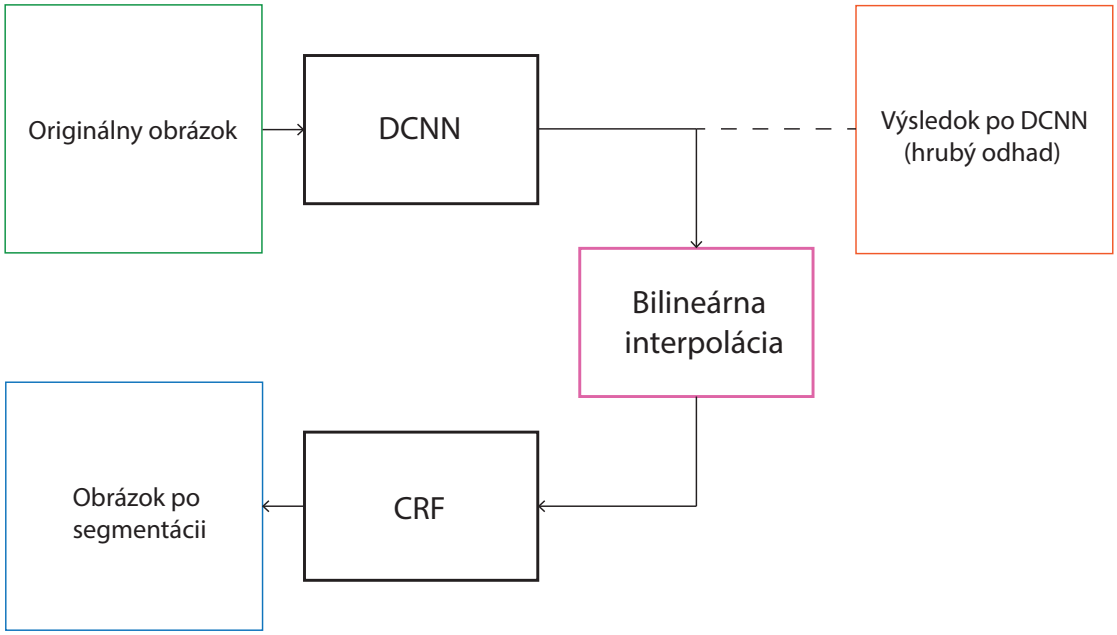
Obr. 3.5: Algoritmus dier v 1-D, pričom veľkosť filtru je 3, veľkosť vstupného kroku je 2 a veľkosť výstupného kroku je 1



Obr. 3.6: Príklad štandardnej konvolúcie v prípade výberu (extrakcie) príznakov s riedkym rozdelením a algoritmu dier v prípade extrakcie príznakov s hustým rozdelením

Aplikácia algoritmu dier bola realizovaná prostredníctvom `im2col` funkcie v rámci frameworku Caffe, ktorého opis je uvedený v podkapitole 4.1. Funkcia štandardne vykonáva konvertovanie mnohokanálovej mapy na vektorizované políčka a bola doplnená schopnosťou redšieho výberu z tejto základnej mapy, čím sa aplikácia stáva všeobecne platná a efektívna bez potreby zavedenia ďalšej aproximácie pri akejkoľvek zvolenej rýchlosti nadvzorkovania.

Základný model VGG-16, ktorý je inicializačným modelom metódy Deeplab produkuje vo výsledku tisíc kategórií (tried), tento počet je v modeli metódy Deeplab zredukovaný na dvadsaťjeden. Stratová funkcia je súčtom vyjadrení krížnej entropie pre všetky priestorové pozície v rámci výstupnej mapy, ktorá je však nadvzorkovaná osemnásobne v porovnaní s originálnym obrázkom na vstupe. Všetky pozície a triedy sú rovnomerne vážené v rámci celkovej funkcie strát [11].



Obr. 3.7: Bloková schéma siete pre sémantickú segmentáciu obrazu metódou Deeplab

Z blokovej schémy siete Deeplab pre sémantickú segmentáciu na obr. 3.7 je možné vidieť uplatnenie bilineárnej interpolácie pre zvýšenie rozlíšenia hrubého výstupu vyhodnotenej mapy tried - konkrétne o faktor osem pričom výpočtové náklady sú zanedbateľné. Využitie interpolácie je v tomto prípade možné vďaka algoritmu dier, resp. vďaka jemnejšiemu výsledku zmienenej mapy tried v porovnaní s CNN, ktoré tvoria základ pri segmentácii metódou CRF-RNN. U CRF-RNN sa algoritmus dier neimplementuje a výsledky sú omnoho hrubšie, inými slovami horšie rozpoznateľné a nadvzorkované o faktor tridsaťdva na výstupe CNN. S tým je spojená nutnosť

použitia naučených nadvzorkovacích vrstiev, čím sa podstatne zvyšuje zložitosť siete a čas nutný pre učenie systému [11].

3.2.2 Dodatočné vyhľadenie pomocou CRF

Podstatným rozdielom medzi metódou Deeplab a väčšinou existujúcich metód, ktoré využívajú dodatočné vyhľadenie výsledkov pomocou pravdepodobnostných modelov je fakt, že tieto metódy pracujú na úrovni superpixelov, ktoré považujú za uzly pre lokálne náhodne polia spolu s rezmi grafov pre konečnú dedukciu. Takýto výsledok však može byť značne limitovaný práve pre chyby v superpixelových výpočtoch. U metódy Deeplab sa zaobchádza s každým pixelom ako s CRF uzlom, využívajúc závislosti na dlhých dosahoch a finálnu dedukciu po CRF. Výsledkom tohto postupu je priama optimalizácia hodnotovej funkcie na základe DCNN. Podrobnejší popis podmienených náhodných polí je možné nájsť v podkapitole 3.1.1. Metóda Deeplab bola zvolená pre vlastné trénovanie v rámci práce, detaily o možnosti trénovania sú uvedené v podkapitole 4.2.2.

4 IMPLEMENTÁCIA

Pre úspešnú implementáciu je potrebné zvoliť vhodné prostredie pre využitie hlbokého učenia. V kapitole sú popísané niektoré z nich a ďalej konkrétne prostredie zvolené pre realizáciu trénovania a testovania sémantickej segmentácie v tejto práci. Z pohľadu vybraného prostredia sú potom zmienené niektoré základné súčasti a požiadavky pre ich správnu funkčnosť. V kapitole sú taktiež pojednávané základne možnosti trénovania, forma sietí, predtrénované ale i vzniknuté modely. Kapitolu uzatvára popis metriky, podľa ktorej sa v práci hodnotí presnosť trénovaných modelov.

4.1 Prostredia pre implementáciu hlbokého učenia

Pre využitie vlastností hlbokého učenia existuje rada prostredí, v ktorých sa dosahuje značný rozvoj v efektivite, rýchlosti trénovania a univerzálnosti využitia. V tejto podkapitole budú uvedené niektoré využívané prostredia, ich vlastnosti či zameranie a podporované rozhrania programovacích jazykov. Výkonnejšie výpočty pri riešení úloh s využitím hlbokého učenia na grafickom procesore (GPU – Graphics Processing Unit) sú dosiahnuté vďaka podpore architektúry CUDA.

Caffe

Caffe framework poskytuje výskumníkom a odborníkom v oblasti počítačového videnia jednoduché a prispôsobiteľné prostredie pre najlepšie výsledky pri učení a aplikovaní všeobecne zameraných CNN ako i iných modelov využívaných v oblasti hlbokého učenia. Framework spadá pod BSD licenciu a je preto voľne dostupný. Podporuje rozhranie jazyka C, C++, Python, Matlab a rozhranie príkazového riadku. Caffe bol vyvinutý skupinou BVLC (Berkeley Vision and Learning Center) a v súčasnosti je ďalej vyvíjaný i komunitou prispievateľov, priaznivcov tohto prostredia [33].

CNTV

Skratka vychádza z názvu „The Microsoft Cognitive Toolkit“, prostredie bolo v minulosti známe pod označením CNTK a predstavuje zjednotený súbor nástrojov pre hlboké učenie. Vývoj tohto prostredia vznikol na základe výskumu spoločnosti Microsoft a jeho cieľom je uľahčiť učenie a kombinovanie populárnych modelov naprieč

rôznymi grafickými kartami a servermi. Podporuje rozhranie jazyka C#, C++, Python a rozhranie príkazového riadku. Tento nástroj implementuje vysoko efektívne CNN a RNN trénované pre rozpoznanie reči či textových a obrazových dát [34].

TensorFlow

TensorFlow je softvérová knižnica pre numerické výpočty pomocou grafov toku dát, vyvinutý výskumnou organizáciou strojovej inteligencie spoločnosti Google. Jednotlivé uzly v grafe predstavujú matematické operácie a hrany predstavujú viacdimenzionálne dátove polia. Podporuje rozhranie jazyka C++ a Python [35].

Theano

Theano je vyvíjaný Univerzitou v Montreale a predstavuje knižnicu jazyka Python. Využíva sa ako kompilátor matematických výrazov, ktorý efektívne určuje, optimalizuje a vyhodnocuje matematické vyjadrenia zahrňujúce viacrozmerné polia. Podporuje rozhranie jazyka Python [36].

Torch

Torch je vedecky zameraný výpočetný framework, ktorý poskytuje širokú podporu pre algoritmy hlbokého strojového učenia. Je založený na programovacom jazyku Lua a používa skriptovací jazyk LuaJIT. Podporuje rozhranie jazyka C, C++, Lua [37].

MXnet

Framework s využitím hlbokého učenia navrhnutý pre efektivitu a zároveň flexibilitu, ktorá umožňuje kombináciu rôznych štýlov symbolického programovania spolu s imperatívnym programovaním, od cloud infraštruktúr až po mobilné zariadenia a to s účeľom maximalizácie produktivity a efektívnosti. Podporuje rozhranie jazyka Python, R, C++, Julia, Scala [38].

Chainer

Framework pre hlboké učenie, ktorý je navrhnutý na princípe, kedy si individuálne vrstvy (stavebné bloky) pamätajú svojich predkov. V porovnaní s inými frameworkami, Chainer umožňuje zmenu siete tzv. „za behu“, čo zároveň znamená použitie ľubovoľných príkazov pre riadenie toku. Podporuje rozhranie jazyka Python [39].

Keras

Keras je minimalistická, vysoko modulárna knižnica pre neurónové siete, založená na jazyku Python. Jej praktické využitie je na základe prostredia TensorFlow alebo Theano. Cieľom vývoja tejto knižnice bolo sprístupnenie vykonávania rýchlych experimentácií v oblasti hlbokého učenia. Podporuje rozhranie jazyka Python [40].

DL4J

Deeplearning4j je knižnica pre hlboké učenie, vyvinutá pre jazyk Java a Java virtuálne prostredie. Framework podporuje širokú škálu algoritmov a modelov s využitím hlbokého učenia. Pre distribuované uchovávanie a spracovávanie veľkého toku dát sú integrované frameworky Hadoop a Spark. Hlavnou prednosťou je jednoduchosť pri prototypovaní modelov. DL4J je voľne dostupným softvérom pod licenciou Apache License 2.0., v komerčnej oblasti je podporovaný skupinou Skymind. Podporuje rozhranie jazyka Java, Scala [41].

4.1.1 Aplikácia konkrétneho prostredia

Pre úspešnú realizáciu sémantickej segmentácie s využitím hlbokého učenia bol v práci použitý framework Caffe. Upravená verzia použitého prostredia Caffe pre metódu CRF-RNN spolu s ostatnými potrebnými súbormi pre implementáciu segmentácie vrátane CRF, je voľne dostupná z [42]. V metóde Deeplab sa využíva staršia verzia prostredia Caffe, ktorá je voľne dostupná z [43]. Pre správnu funkciu je potrebné nainštalovať a nakonfigurovať niekoľko knižníc a ostatných súčastí.

V práci sa využívajú knižnice `protobuf, leveldb/lmdb, snappy, glog, hdf5, opencv, gflags`. Nutnosťou je i inštalácia sady knižníc `boost` využívaných pre C++ a podľa vlastného výberu je nutné použiť jednu z knižníc `Atlas/MKL/OpenBlass` aplikovaných pre podporu základných algebraických subprogramov.

Dôležitou súčasťou určenou pre výpočty na grafickej karte je paralélna výpočetná platforma a programovací model CUDA od spoločnosti NVIDIA. Využitím tejto platformy je dosiahnutý lepší výkon a rýchlosť pri riešení úloh a výpočtov na grafickej karte. Aplikáciu konkrétnej verzie je potrebné prispôsobiť konfigurácii používaného systému, rovnako je potrebné zvýšiť pozornosť pri kompatibilite s ovládačom GPU. V dobe písania tejto práce sa odporúča verzia knižnice 7 (a vyššie) s najnovším ovládačom. Cestu pre CUDA je taktiež potrebné definovať v súbore `.bashrc`.

Pred kompiláciou je potrebné upraviť konfiguračný súbor `Makefile.config`. Tento súbor sa používa pre nastavenie ciest ku knižniciam a okrem iného i pre výber jednotky, ktorá bude použitá pre výpočet a teda grafická karta či procesor.

Po úspešnom nastavení prostredia je potrebné vykonať spomínanú kompiláciu a testami overiť funkčnosť, konkrétne v kroku `make all, make test, make runtest`. Ak testy prebehnú úpešne, je možné pokračovať k nasledujúcej konfigurácii rozhrania Python alebo Matlab, opäť je možnosť voľby podľa riešiteľa. V práci sa uplatňuje Python a dokonfigurácia pre Caffe, tzv. „pycaffe“.

Ďalším voliteľným prvkom, ktorý je možné využiť v rámci frameworku je knižnica cuDNN pre hlboké neurónové siete, určená pre GPU od spoločnosti NVIDIA. Vyvinutá bola so zámerom zrýchlenia výpočtov a odľahčenie nákladov operácií Caffe. Pre využitie cuDNN je pred kompiláciou potrebná konfigurácia súboru `Makefile.config` pre nastavenie cesty k umiestneniu cuDNN v systéme. Táto knižnica bola súčasťou konfigurácie pri úspešnom testovaní modelu CRF-RNN. V prípade Deeplab nebola táto knižnica využitá.

Inštalácia a konfigurácia daného frameworku bola sprevádzaná s veľkým množstvom problémov a chybových hlášok, ktoré boli väčšinou spôsobené nesprávanymi a nekompatibilnými verziami, chýbajúcimi definíciami premenných či chybami v kóde.

Ďalšie podrobnosti o nasadení prostredia Caffe je možné nájsť priamo na stránke vývojárov skupiny BVLC [44].

4.2 Trénovanie neurónovej siete

Táto podkapitola sa zaoberá princípom trénovania na základe konvolučnej neurónovej siete s cieľom segmentácie na úrovni pixelov. Popísané sú obe zvolené metódy ako CRF-RNN tak i Deeplab v praktickej rovine. V texte je vysvetlená príslušná úprava siete a opis modelu, obsah datasetu a základné rozdelenie trénovacej množiny.

4.2.1 Metóda CRF-RNN

Postup pri trénovaní nového modelu na princípe metódy CRF-RNN by mal byť v prvom kroku založený na trénovaní jednoduchými FCN-8 bez použitia ďalších mechanizmov pre následné spracovanie, v tomto prípade bez realizácie CRF. Po ukončení trénovania na základe siete FCN, nastáva aplikácia metódy CRF-RNN v plnom rozsahu a trénovanie celou sieťou. Aj napriek snahe trénovania a konfigurácie siete na základe metódy CRF-RNN sa v práci nepodarilo dosiahnuť odpovedajúcich výsledkov v porovnaní so zverejnenými výsledkami v [18] a to najmä v prípade trénovania s iným počtom výsledných tried. Tento problém pripisujem nesprávnej inicializácie dekonvolučnej vrstvy v sieti.

Ďalší postup v práci bol založený na štúdiu možných variánt segmentácie na úrovni pixelov s využitím grafických modelov. Jednou z možných metód segmentácie

je Deeplab, ktorej sa venuje i podkapitola 3.2 a jej základom je využitie DCNN s následným spracovaním pomocou podmienených náhodných polí.

Pre overenie vhodnosti použitia metódy Deeplab pre sémantickú segmentáciu sa výsledky po trénovaní touto metódou porovnajú s výsledkami predtrénovaného modelu na princípe CRF-RNN. Opis predtrénovaného modelu s názvom COCO-VOC je potom následovný: model bol realizovaný pomocou frameworku Caffe a bol založený na postupnom trénovaní FCN sieťami. Prvým krokom sa stalo trénovanie prostej FCN-32 siete na sade dát COCO VOC 2014 [1]. Následne sa vybudovala sieť FCN-8 (teda osem-pixelové (osemkrát nadvzorkované) predikcie na krok siete) s naučenými váhami a v poslednej časti prebehlo trénovanie celou CRF-RNN sieťou na základe sade trénovacích dát PASCAL VOC 2012 [45]. Trénovacích dát (obrázkov) je tak celkovo 77784.

Pre elimináciu prekročenia kapacity GPU sa počas trénovania uplatnilo spracovávanie práve jedného obrázku na dávku [2]. Toto nastavenie platí v prípade predtrénovaného modelu COCO-VOC ako i pre vlastné trénovanie metódou Deeplab. V prípade konfigurácie vrstvy CRF je počet iterácií pri testovaní zvolených na hodnotu desať. Počet iterácií pri tréningu odpovedalo hodnote päť, nakoľko pri použití desiatich iterácií dochádza k zníženiu presnosti, za čo je zodpovedný efekt miznúceho gradientu chybovej funkcie pri spätnom šírení. Porovnanie výsledkov segmentácie po trénovaní FCN-8 bez aplikácie CRF, s aplikáciou CRF ako dodatočným spracovaním a modelom CRF-RNN trénovaným celou sieťou je možné nájsť v [18]. V publikácii sa udáva i hodnota koeficientu IoU (Intersection over Union) pre hodnotenie modelu trénovaného na rôznych druhoch databáz. Použitie odlišných váh pre odlišné triedy vedie k zlepšeniu presnosti a taktiež využitie asymetrickej kompatibilnej transformácie zdokonaluje celkový výkon [18].

Úspech a detaily z vlastného trénovania modelu na základe metódy Deeplab je možné nájsť v časti práce, ktorá pojednáva o dosiahnutých výsledkoch 6. Obrázky pre porovnanie úspešnosti segmentácie metódou CRF-RNN a metódou Deeplab sú potom dostupné v prílohe k textu práce A.3 na obr. A.5.

4.2.2 Metóda Deeplab

Vlastné trénovanie bolo vykonané na princípe metódy Deeplab a v práci sa táto metóda využila pre dva rôzne datasety. Pre správne nastavenie parametrov trénovania a dosiahnutia čo najlepších výsledkov je potrebné adekvátne prispôsobiť konfiguračný súbor, typicky pomenovaný ako `solver`. Konfiguračný súbor je typu prototxt a je implementovaný v prostredí Caffe. Definované parametre v tomto súbore sú potom

[1] http://mscoco.org

[2] z anglického významu batch

dôležité najmä z pohľadu optimalizácie procesu trénovania pre dosiahnutie čo najvyššej presnosti vo výslednom modeli. Medzi parametre, ktoré sa v našom prípade prispôsobovali a upravovali pre optimalizáciu sú: `base_lr`, ktorý určuje počiatočné tempo učenia, definuje sa číselnou hodnotou a pri trénovaní sémantickej segmentácie je jeho hodnota väčšinou vyjadrená v desatinných číslach napr. 0,01. `lr_policy` indikuje akým spôsobom sa tempo učenia mení počas trénovania, v tomto prípade sa vyjadruje jeho hodnota príslušným kľúčovým slovom vyjadreným v úvodzovkách. V práci je tento parameter vo všetkých prípadoch definovaný kľúčovým slovom `step`, ktorým sa zníži tempo učenia v závislosti na koeficiente `gamma`. Koeficient `gamma` potom opäť číselne vyjadruje hodnotu, o ktorú sa má tempo učenia zmeniť, po dosiahnutí určitého počtu iterácií vyjadrených parametrom `stepsize`. Po dosiahnutí zvoleného počtu iterácií sa trénuje s novým tempom učenia. Celkový počet iterácií určuje parameter `max_iter`. `Momentum` je stanovený číselnou hodnotou vyjadrujúcou mieru preučenia váh z inicializačného modelu, `weight_decay` je faktor pre penalizáciu veľkých váh a `solver_mode` sa uplatňuje pri výbere platformy CPU / GPU. Vďaka číselnej hodnote parametra `snapshot` sa volí pravidelnosť pre ukladanie nového modelu, ktorého názov je možné definovať pomocou ďalšieho parametru `snapshot_prefix`. Diagram vzniknutého modelu a jeho jednotlivé vrstvy je možné nájsť v prílohe k textu práce na obr. A.1.

Štandardne sa ako inicializačný model pre experimenty založené na metóde Deeplab využíva `vgg16_128.caffemodel`, ktorý má veľkosť filtru 4x4 pre prvú plne pospájanú vrstvu a filtre typicky 4096. V našom prípade bol ako inicializačný model zvolený `vgg16_20M.caffemodel` [43]. Model sa vyznačuje použitím veľkosti filtru 3x3 pre prvú plne pospájanú vrstvu a aplikovaním 1024 filtrov. Táto vlasnosť predstavuje veľkú výhodu v prípade obmedzenej kapacity pre realizáciu výpočtov ako to bolo v prípade využitia grafickej karty NVIDIA GTX 980 s pamäťou 4 GB. Pred trénovaním pomocou metódy CRF-RNN s počtom filtrov 4096 bolo nutné zmenšovať rozmery obrázkov až na 300x300 pixelov. Takýto druh limitácie je možné eliminovať práve použitím modelu `vgg16_20M` u metódy Deeplab.

Okrem inicializačných modelov je možné v oficiálnom archíve projektu Deeplab [43] nájsť i konfiguračné súbory pre trénovanie a testovanie, prípadne iné pomocné skripty. Ako dobrá pomôcka môže poslúžiť i depozitár [46] najmä z pohľadu konvertovania formátov, tu je ale podstatná ostražitosť pre nutné odstránenie chýb v kódových častiach a celkové prispôsobenie systému.

4.2.3 Hodnotenie presnosti

Pre vyhodnotenie úspešnosti modelu sa v práci využívajú metrika IoU. Táto metrika tvorí kľúčovú podstatu pre hodnotenie modelu v oblasti sémantickej segmentácie

nakoľko zohľadňuje priemer úspešnosti na všetkých správne i nesprávne priradených pixelov v porovnaní s ground-truth a to pre všetky triedy.

IoU [3] je vyjadrená vzťahom:

$$\frac{1}{n_{cl}} \sum_{i} \frac{n_{ii}}{(t_i + \sum_j n_{ji} - n_{ii})}. \tag{4.1}$$

Kedy n_{cl} vyjadruje počet tried zahrnutých v ground-truth, n_{ij} určuje počet pixelov triedy i predpovedaných do triedy j, t_i je celkový počet pixelov triedy i v ground-truth [9].

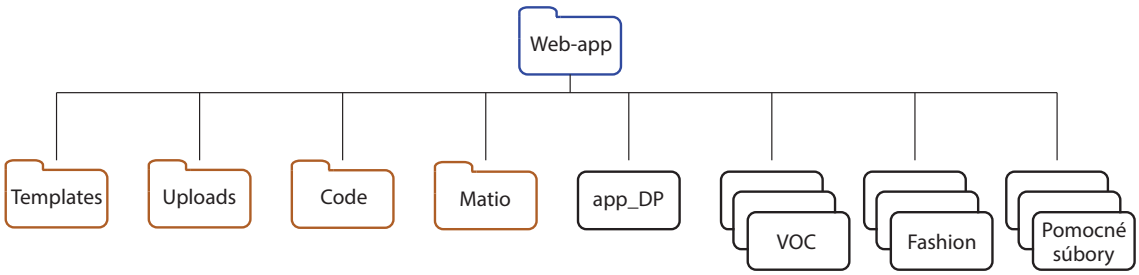
[3] Z anglického významu mean Intersection over Union

5 NÁVRH UŽÍVATEĽSKÉHO ROZHRANIA

V následujúcej kapitole je popísané navrhnuté užívateľské rozhranie pre realizáciu sémantickej segmentácie s využitím hlbokého učenia na základe natrénovaných modelov. Vysvetlený je princíp z pohľadu architektúry a adresárovej štruktúry, ďalej je opísaný vzhľad stránok spolu s popisom interakcie s navrhnutým prostredím. Hlavná časť pre implementáciu – trénovanie i testovanie je navrhnuté v jazyku Python a z dôvodu jednoduchej integrácie je webové rozhranie realizované taktiež prostredníctvom jazyka Python, tzv. Python Flask [1] web server.

5.1 Architektúra rozhrania

Ako prvé je v rámci architektúry rozhrania načrtnuté spracovanie adresárovej štruktúry na obr. 5.1. Koreňový adresár s názvom `Web-app` obsahuje všetky dôležité súčasti pre správnu činnosť rozhrania.



Obr. 5.1: Štruktúra koreňového adresára s farebným odlíšením priečinkov a súborov

Obsah koreňového adresára potom tvorí: súbor pre spustenie webového rozhrania s názvom `app_DP.py`, ktorý predstavuje kompletné vypracovanie navrhnutého rozhrania v jazyku Python. Priečinok `templates`, ktorého súčasťou sú HTML šablóny s využitím frameworku Bootstrap [2]. Ďalej sa tu nachádza priečinok `uploads`, do ktorého sa ukladajú obrázky pri výbere i výsledné obrázky po segmentácii. Taktiež obsahuje obrázky tried, ktoré sa využívajú v úvodnej a poslednej stránke rozhrania. Viac o vzhľade rozhrania je možné nájsť v kap. 5.2. Priečinok `code` obsahuje kompilovanú verziu Caffe a všetky jej potrebné súčasti ako i súbory CRF. Posledným priečinkom je `matio` pre prácu s binárnymi MATLAB `mat` súbormi.

[1] http://flask.pocoo.org/

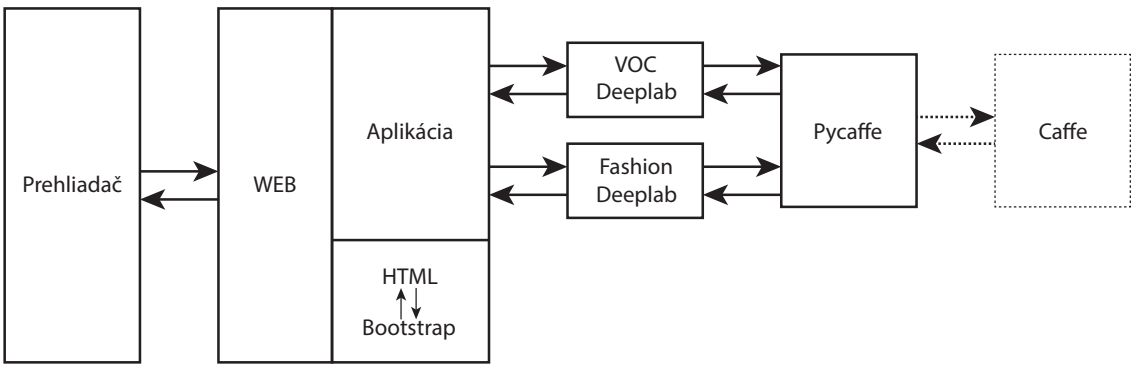
[2] Sada nástrojov pre tvorbu webu a webových aplikácií, obsahuje návrhárske šablóny založené na HTML a CSS

Princíp segmentácie v pozadí

Pre vykonanie segmentácie sa parametre definované v `app_DP.py` predajú súboru `názov_modelu_deeplab.py`, ktorý je zodpovedný za realizáciu segmentácie s podporou ostatných pomocných súborov. Pre každý model potom existuje samostatný súbor a ako `názov_modelu` môže v našom prípade figurovať VOC alebo Fashion. Definované sú nasledujúce parametre:

- GPU id
- sieť pre realizáciu segementácie (prototxt súbor)
- natrénovaný model
- vstupný obrázok

`GPU id` umožňuje výber platformy CPU/GPU podľa priradeného id (0,1...). `Sieť` opisuje zloženie jednotlivých vrstiev a súbor je typu prototxt. `Natrénovaný model` predstavuje model, ktorý vznikol vlastným trénovaním alebo akýkoľvek iný predtrénovaný model, zvolený pre danú segmentáciu. `Vstupný obrázok` je priamo závislý na výbere užívateľa. Pre zaručenie unikátnosti každého obrázka sa k jeho názvu pridáva reťazec znakov podľa uuid[3] verzie 4. K názvu výstupného obrázka po segmentácii sa pridáva prípona ***-label***. Vstupné obrázky v prípade modelu VOC nepresahujú veľkosť šírky alebo výšky `500 px` a v prípade modelu Fashion `600 px`. Ak je vybraný obrázok väčší, automaticky sa upravuje podľa najväčšej hodnoty šírky / výšky so zachovním aspektu na maximálne definovaný rozmer. Obecný princíp navrhnutého rozhrania je znázornený blokovou schémou na obr. 5.2. Každý zo zmienených para-



Obr. 5.2: Bloková schéma architektúry navrhnutého rozhrania pre segmentáciu

metrov je definovaný v `app_DP.py` a podstatnou výhodou rozhrania je fakt, že nenáročnou zmenou parametrov je možné realizovať segmentáciu na základe akejkoľvek inej siete pri použití príslušného natrénovaného modelu a pri zachovaní Caffe verzie.

[3]Universally unique identifier

Súčasná štruktúra je teda sústredená v jednom koreňovom adresári pre jednoduchú manipuláciu so všetkými využívanými súbormi. V prípade potreby, napr. použitia odlišnej verzie Caffe alebo jej rozdielneho umiestnenia v systéme, je potrebné adekvátne uspôsobiť všetky cesty podľa umiestnenia zvolenej verzie v systéme. Opísaným princípom je možné tvrdiť, že navrhnuté rozhranie má univerzálne využitie.

5.2 Vzhľad rozhrania

Rozhranie je dostupné na adrese `0.0.0.0:5000` a užívateľ je týmto spôsobom schopný integrovať s aplikáciou. Vzhľadom na to, že nami testovaný systém bol školský server, jedná sa konkrétne o adresu `gravy.utko.feec.vutbr.cz:5000`. Ako bolo spomenuté v časti o architektúre rozhrania, vzhľad stránky je tvorený pomocou HTML šablón s využitím frameworku Bootstrap.

Úvodná stránka

Úvodnú stránku, tak ako i všetky ostatné tvorí nadpis zložený z názvu témy tejto diplomovej práce a teda: `Segmentácia obrazu s využitím hlbokého učenia`. Pod názvom sú umiestnené dva panely, jeden pre možnosť segmentácie na základe natrénovaného modelu VOC a druhý pre možnosť segmentácie na základe natrénovaného modelu Fashion. Vzhľad úvodnej stránky je možné nájsť v prílohe A.2.1 na obr. A.2.

Výber obrázka pre segmentáciu je realizovateľný dvomi spôsobmi:

1. Výber obrázka uloženého priamo v počítači, resp. na disku
2. Výber obrázka na základe URL adresy

Takúto možnosť poskytujú oba spomínané panely. Pre potvrdenie výberu obrázka a prechod na druhú stránku je potrebné už len kliknúť na možnosť `nahraj`. Okrem spomínaného výberu je na úvodnej stránke vyznačená použitá metóda pre segmentáciu, t. j. `Deeplab`. Poslednou súčasťou sú dva obrázky opisujúce všetky triedy, ktoré je možné rozpoznať pre každý model. Pochopiteľne sú tieto zobrazené triedy odlišné v počte, druhu i farebnom prevedení zvlášť pre model VOC a pre model Fashion.

Vybraný obrázok

Po úspešnom výbere obrázka je užívateľ presmerovaný na druhú stránku, ktorej súčasť tvorí opäť nadpis, tentokrát je panel z úvodnej stránky zjednotený a zobrazuje sa iba názov zvoleného modelu viď obr. A.3 nachádzajúci sa v prílohe A.2.2. Hlavnou podstatou tejto stránky je zobrazenie obrázka pre segmentovanie v náväznosti na výber v úvodnej stránke. Pod obrázkom sa nachádza tlačítko `segmentuj`, ktorého

význam vysvetľuje text pod ním. Kliknutím na tlačítko segmentuj sa v pozadí spustí segmentácia a užívateľ v tomto prípade čaká na realizáciu segmentovania a zobrazenie výsledku. V prípade však, že sa užívateľ rozhodne segmentovať iný obrázok ako pôvodne zvolený v úvode, má možnosť vrátiť sa jednoduchým spôsobom – kliknutím na text `Klikni sem pre segmentovanie iného obrázka`. Týmto spôsobom je užívateľ opäť presmerovaný na úvodnú stránku.

Zobrazenie výsledku po segmentácii

Ako bolo zmienené v predchádzajúcom kroku, pri výbere obrázka, jeho nahratí a spustení segmentácie sa v pozadí vykoná segmentácia. Celkový čas pre segmentáciu, predanie obrázka oboma smermi i následné zobrazenie vysegmentovaného obrázka netrvá dlhšie ako šesť sekúnd. Stránku so zobrazeným výsledkom tvorí štandartne nadpis, názov zvoleného modelu a dva obrázky. V ľavej časi sa zobrazuje originálny obrázok, ktorý bol určený pre segmentovanie, v pravej časti sa zobrazuje výsledok po segmentácii. Nad obrázkami je opäť vysvetlivka v podobe textu pre rozlíšenie originálneho a vysegmentovaného obrázka. V spodnej časti stránky sa nacháda zoznam tried i s príslušným farebným rozlíšením pre každú triedu, vďaka čomu dokáže užívateľ posúdiť správnosť a presnosť rozponaného výsledku. Ak chce užívateľ pokračovať v segmentácii iného obrázka má opäť možnosť vrátiť sa jednoduchým spôsobom – kliknutím na text `Klikni sem pre segmentovanie iného obrázka`, týmto spôsobom je užívateľ presmerovaný na úvodnú stránku. Príklad vzhľadu stránky so zobrazeným výsledkom po segmentácií je možné vidieť na obr. A.4 v rámci prílohy A.2.3.

6 VÝSLEDKY

Úvod kapitoly sa venuje opisu systému, na ktorom bola segmentácia realizovaná. Ďalej sú uvedené detaily o použitých databázach, z ktorých boli čerpané obrázky pre trénovanie vlastných modelov. V rámci výsledkov trénovania sú vysvetlené jednotlivé nastavenia parametrov a dosiahnuté výsledky zobrazené v grafoch či v podobe prehľadových tabuliek, doplnené o niekoľko ukážok výsledných segmentácií.

V tab. 6.1 sú uvedené niektoré technické parametre ako softvérovej časti tak i hardvérovej časti systému. V práci boli využívané dve rôzne grafické karty, pre ďalšie použitie v texte sa grafická karta `NVIDIA GEFORCE GTX 980` bude označovať ako `GPU 1` a grafická karta `NVIDIA GEFORCE GTX 1080 Ti` ako `GPU 0`. V rámci doplňujúcich informácií je vhodné uviesť, že trénovanie a testovanie na GPU 1 prebiehalo s podporou CUDA verzie 7 a v prípade GPU 0 s podporou CUDA verzie 8.

Tab. 6.1: Technické parametre systému, na ktorom bola realizovaná segmentácia

Operačný systém	Ubuntu 14.04.5 LTS
Procesor	2x Intel(R) Xeon(R) CPU E5-2640 v2 @ 2.00 GHz
Pamäť RAM	64 GB
Grafická karta (GPU 1)	NVIDIA GEFORCE GTX 980 4 GB | 2048 jadier CUDA
Grafická karta (GPU 0)	NVIDIA GEFORCE GTX 1080 Ti 11 GB | 3584 jadier CUDA
Verzia Pythonu	Python 2.7.12 :: Anaconda 4.2.0 (64-bit)

6.1 Použité databázy obrazových dát

Databázy obrazových dát sú v oblasti segmentácie zhotovované za účeľom trénovania alebo testovania nového modelu a tvoria vstupné dáta neurónovej siete. V počiatku práce bol vytváraný vlastný dataset pre segmentáciu oblečenia. To sa neskôr ukázalo ako časovo veľmi náročné a pre účely trénovania tak bol zvolený kvalitnejší dataset s dostatočným množstvom anotovaných obrázkov. V práci sa tak celkovo uplatnili dva rôzne datasety pre segmentáciu na úrovni pixelov:

- Dataset VOC 2012 – v práci označovaný ako dataset VOC [45]
- Dataset Humanparsing – v práci označovaný ako dataset Fashion [47]

Obsahom datasetu VOC sú obrázky určené pre segmentáciu objektov a obsahom datasetu Fashion sú obrázky pre segmentáciu častí oblečenia a niektorých doplnkov

módy. Najdôležitejšiu súčasť datasetu tvoria v našom prípade originálne obrázky vo formáte `jpg` a anotované obrázky s indexami tried tzv. „ground-truth“ vo formáte `png`. Ground-truth obrázky sú vnímané neurónovými sieťami ako predloha alebo očakávaný výsledok. Na týchto obrázkoch sú anotované jednotlivé časti objektov, ktoré chceme neurónové siete naučiť a rozpoznať ich tak vo výstupe.

6.1.1 Dataset VOC

Dataset VOC poskytuje možnosť segmentácie s celkovým počtom 20 tried a pozadie. Kompletný prehľad všetkých tried a ich odpovedajúcich indexov pre jednotlivé pixely je uvedený v tab. 6.2 vo formáte `názov triedy:index`. Okrem týchto indexov sa využíva index 0 pre pozadie, resp. nepriradené pixely žiadnej zo zmienených tried. Všetky triedy a ich pridelené farby je možné vidieť na obr. 6.1.

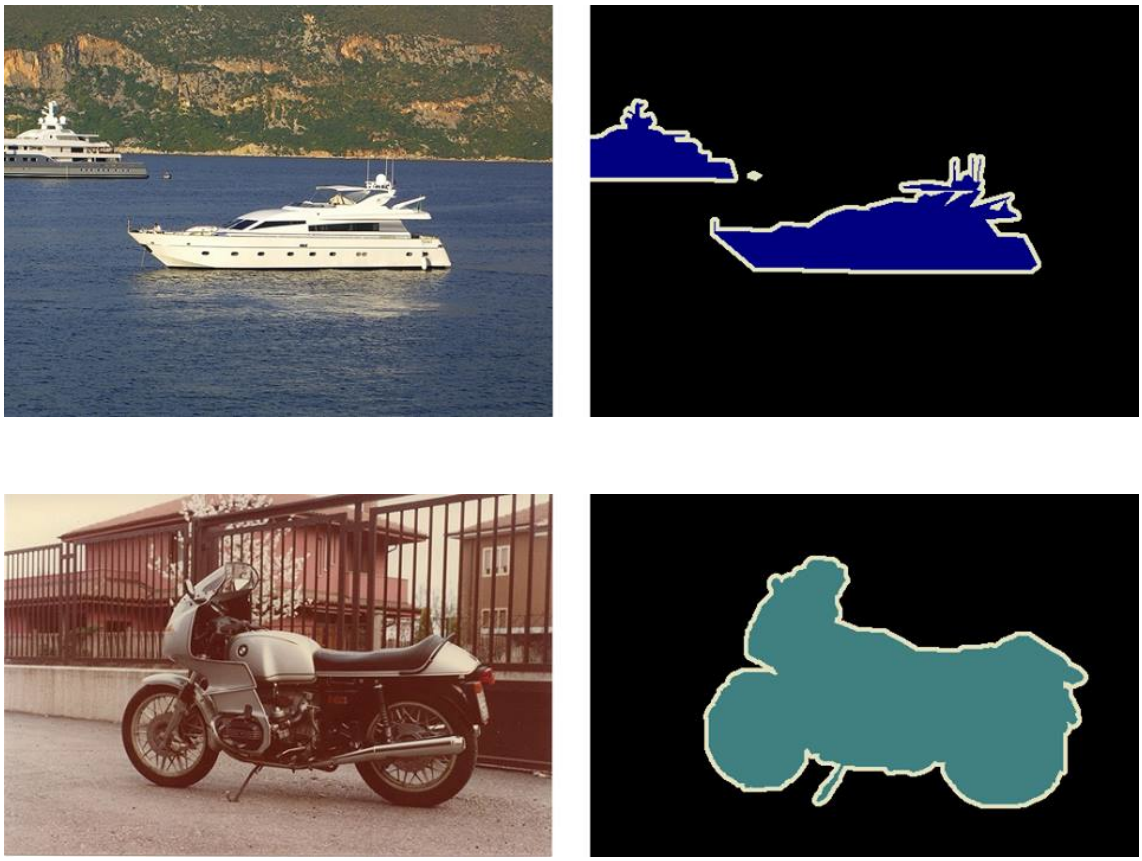
Tab. 6.2: Prehľad tried, ktoré je možné segmentovať v prípade datasetu VOC s príslušným priradením indexu, pozadie má vždy index 0

lietadlo : 1	bicykel : 2	vták : 3	loď : 4	fľaša : 5
autobus : 6	auto : 7	mačka : 8	stolička : 9	krava : 10
stôl : 11	pes : 12	kôň : 13	motorka : 14	osoba : 15
rastlina : 16	ovca : 17	kreslo : 18	vlak : 19	tv/monitor : 20

Obr. 6.1: Farebné odlíšenie tried pre segmentáciu obrazu s využitím datasetu VOC

Príklad originálneho obrázka datasetu VOC a k nemu odpovedajúceho ground-truth obrázka je možné vidieť na obr. 6.2. Dataset je voľne dostupný z oficiálnej stránky VOC [1].

[1] http://host.robots.ox.ac.uk/pascal/VOC/voc2012/



Obr. 6.2: Príklad obrazových dát z datasetu VOC. V ľavej časti sa nachádza originálny obrázok a v pravej časti ground-truth obrázok s anotovaným objektom [45]

6.1.2 Dataset Fashion

Tento dataset poskytuje možnosť segmentácie s celkovým počtom 17 tried a pozadie. Kompletný prehľad všetkých tried a ich odpovedajúcich indexov pre jednotlivé pixely je uvedený v tab. 6.3 vo formáte `názov triedy:index`. Podobne ako u datasetu VOC sa okrem týchto indexov využíva index 0 pre pozadie, resp. nepriradené pixely žiadnej konkrétnej triede. Všetky triedy a ich pridelené farby je možné vidieť na obr. 6.3. Ground-truth obrázky sú v tejto databáze dostupné iba v prevedenom formáte 8-bitového obrázka v odtieňoch šedej. S ohľadom na tento fakt je možné odpovedajúce zakódované indexy tried overiť pomocou akéhokoľvek programu, ktorý dokáže zobraziť informácie o farbách a kanáloch. Pre tento účeľ bol využitý nástroj pre prehliadanie a prácu s obrázkami XnView [2]. Na obr. 6.4 je potom možné vidieť ukážku originálneho obrázka a k nemu odpovedajúceho ground-truth v upravenom formáte s priradeným indexom. Dataset je voľne dostupný z repozitára githubu [3].

[2] http://www.xnview.com

[3] https://github.com/lemondan/HumanParsing-Dataset

Tab. 6.3: Prehľad tried, ktoré je možné segmentovať v prípade datasetu Fashion s príslušným priradením indexu, pozadie má vždy index 0

klobúk : 1	vlasy : 2	okuliare : 3	vrchný odev : 4	sukňa : 5
nohavice : 6	šaty : 7	opasok : 8	ľavá topánka : 9	pravá topánka : 10
tvár : 11	ľavá noha : 12	pravá noha : 13	ľavá ruka : 14	pravá ruka : 15
kabela : 16	šál : 17			

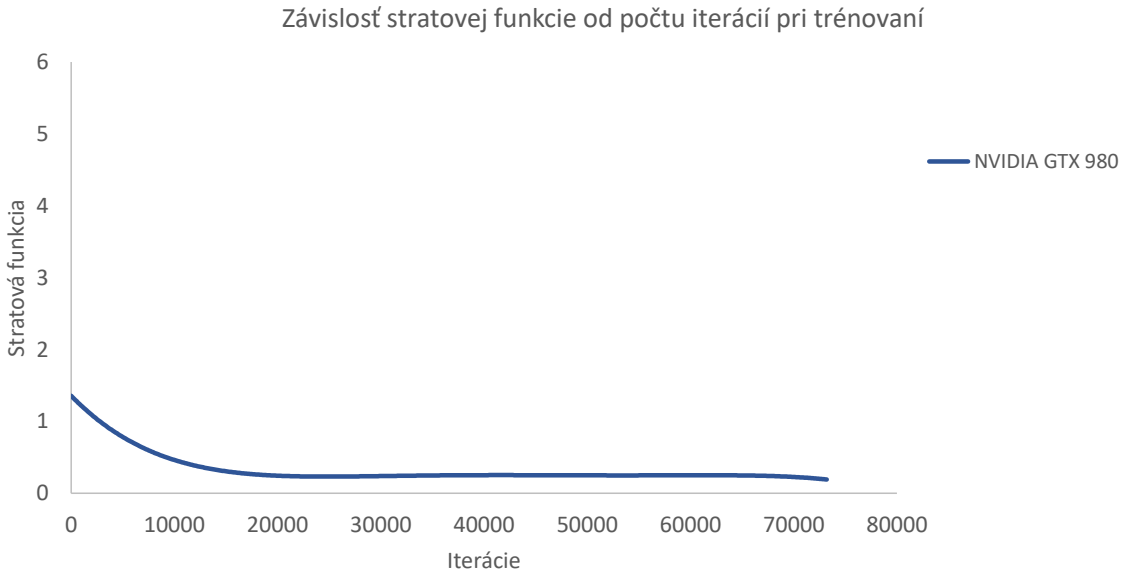
Obr. 6.3: Farebné odlíšenie tried pre segmentáciu obrazu s využitím datasetu Fashion

Obr. 6.4: Príklad obrazových dát, ktoré sú obsahom datasetu Fashion. V ľavej časti sa nachádza originálny obrázok a v pravej časti upravený formát ground-truth obrázka s anotovanými časťami oblečenia [47]

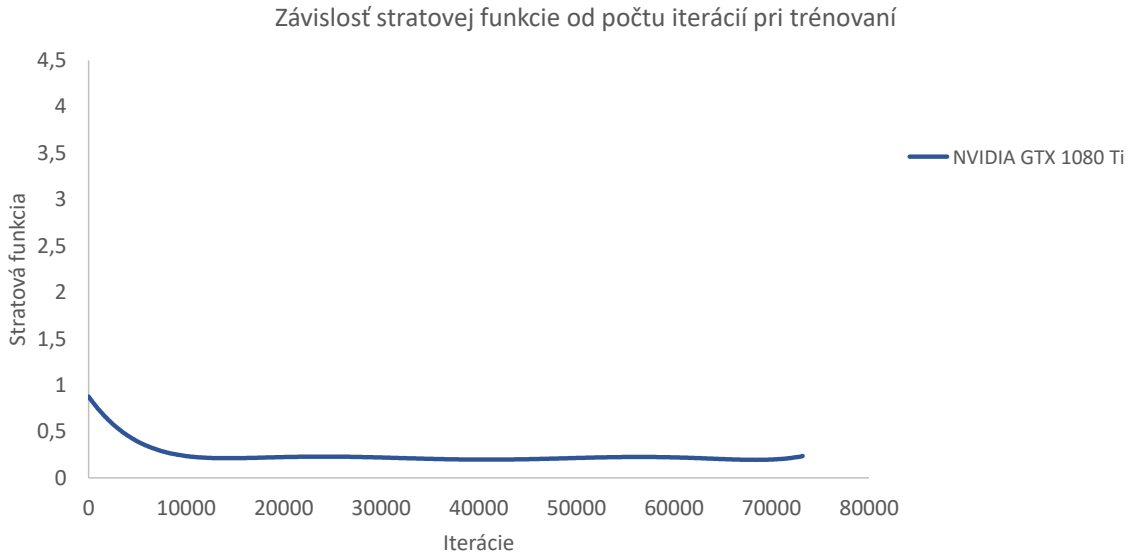
6.2 Výsledky segmentácie s využitím datasetu VOC

`GPU 1` : Pre trénovanie modelu na základe dát z datasetu VOC bol počiatočný učiaci faktor stanovený na hodnotu 0,001. Tento faktor sa počas trénovania automaticky upravoval, po dosiahnutí 10000 iterácií klesla jeho hodnota o jedno desatinné miesto, celkový počet iterácií bol 73200. Rozdelenie datasetu VOC 2012 bolo zvolené v pomere 50:50, čo znamená 1464 obrázkov pre trénovanie a 1449 obrázkov pre testovanie. Vyjadrovací pomer má pri rozdeľovaní dát určitú malú odchýlku, podstata rozdelenia je však zachovaná. Ak je počet trénovacích obrázkov 1464 a počet iterácií je 73200, môžeme hovoriť o celkovom počet 50 epoch. Celkový čas trénovania bol 2 hodiny a 58 minút. Vývoj stratovej funkcie pri trénovaní na GPU 1 je možné vidieť na obr. 6.5.

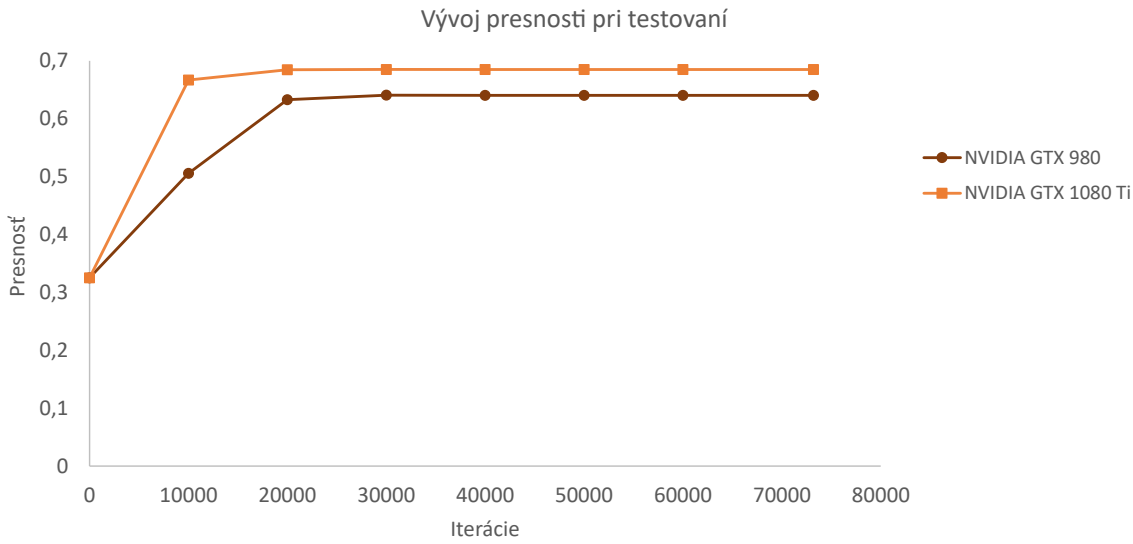
`GPU 0` : Počiatočný učiaci faktor bol upravený na hodnotu 0,0001, nakoľko pri pôvodnej hodnote sa sieť neučila a presnosť počas trénovania dosahovala konštantných hodnôt. Všetky ostatné nastavenia boli zhodné s trénovacími parametrami GPU 1. Celkový čas trénovania bol 1 hodina a 37 minút. Vývoj stratovej funkcie pri trénovaní na GPU 0 je možné vidieť na obr. 6.6. Na základe testovacích dát bola vyhodnotená presnosť trénovaných modelov. Porovnanie dosiahnutej presnosti na oboch dostupných GPU je zobrazené na obr. 6.7. Obsahom tabuľky 6.4 je prehľad vyššie zmienených parametrov a dosiahnutých presností trénovaných modelov s využitím oboch grafických kariet. Ukážka výsledku segmentácie prostredníctvom modelu s najlepšou presnosťou je možné vidieť na obr. 6.8.



Obr. 6.5: Graf trénovania modelu pre sémanticku segmentáciu prostredníctvom GPU NVIDIA GEFORCE GTX 980 a na základe datasetu VOC



Obr. 6.6: Graf trénovania modelu pre sémanticku segmentáciu prostredníctvom GPU NVIDIA GEFORCE GTX 1080 Ti a na základe datasetu VOC



Obr. 6.7: Graf vývoja presnosti počas testovania natrénovaných modelov pre obe GPU. Zdrojom obrazových dát bol dataset VOC

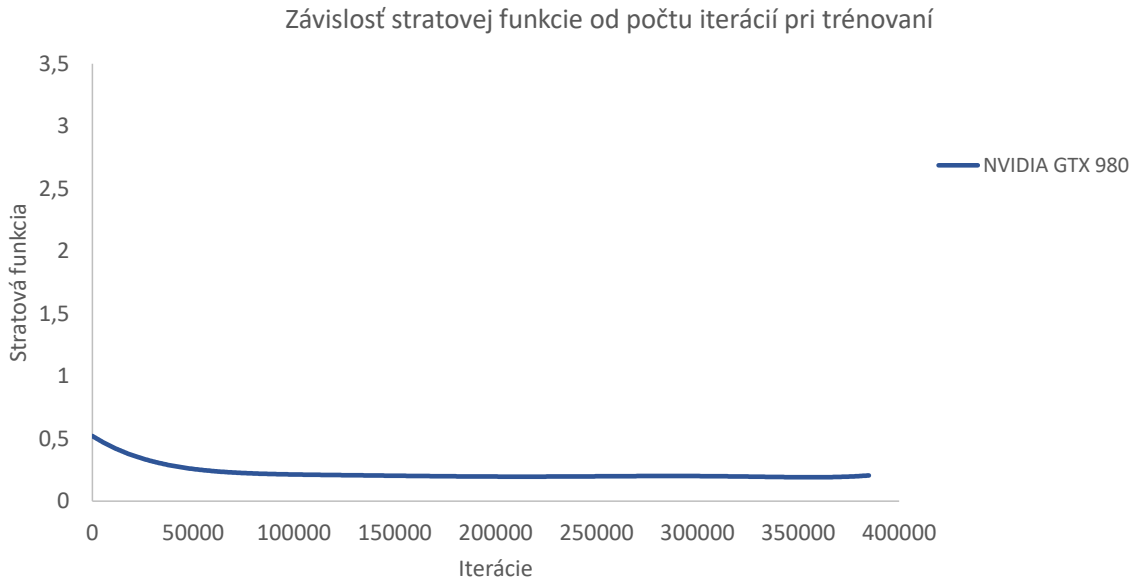
Tab. 6.4: Prehľad zvolených parametroch pre trénovanie a nadobudnutá presnosť na testovacích dátach pomocou datasetu VOC. Veľkosť učiaceho faktoru platí iba do momentu dosiahnutia počtu iterácií uvedených v odpovedajúcom stĺpci

Grafická karta NVIDIA GTX 980							
Počet iterácií	10000	20000	30000	40000	50000	60000	73200
Učiaci faktor	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}
Presnosť IoU [%]	50,58	63,29	64,07	64,03	64,04	64,04	64,04
Grafická karta NVIDIA GTX 1080 Ti							
Počet iterácií	10000	20000	30000	40000	50000	60000	73200
Učiaci faktor	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
Presnosť IoU [%]	66,68	68,47	68,52	68,51	68,51	68,51	68,51

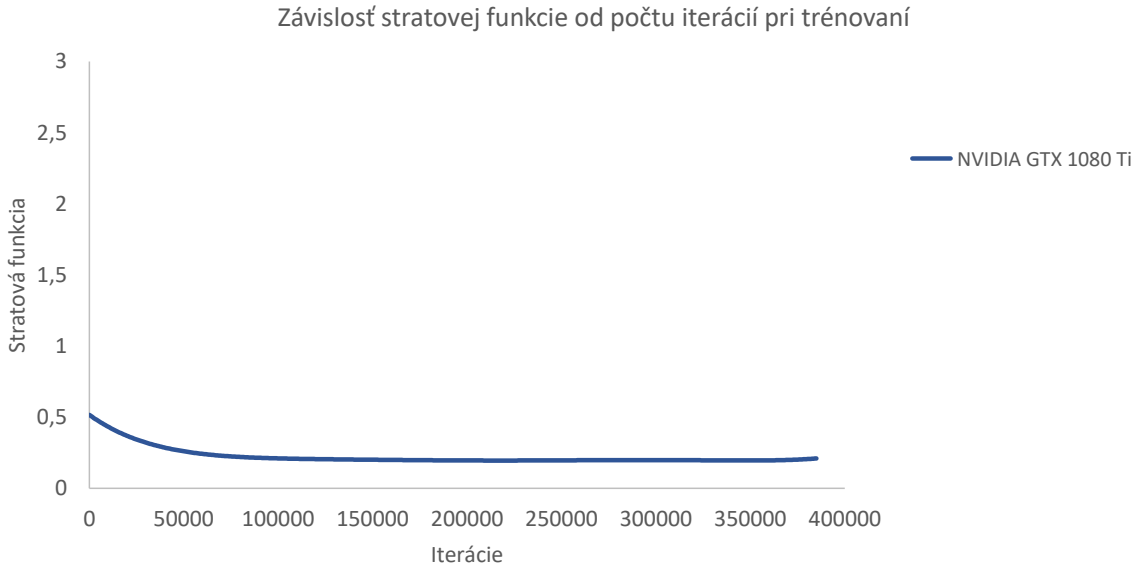
Obr. 6.8: Príklad realizácie segmentácie na natrénovanom modeli. V ľavej časti sa nachádza originálny obrázok a v pravej časti výsledný obrázok po segmentácii [45]

6.3 Výsledky segmentácie s využitím datasetu Fashion

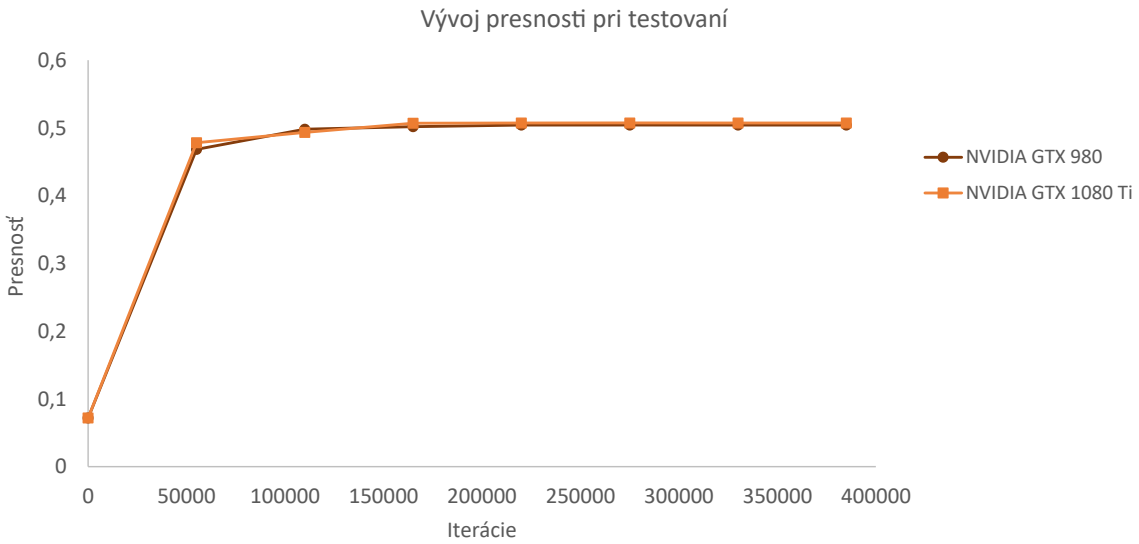
`GPU 1`: Pre trénovanie modelu na základe dát z datasetu Fashion bol počiatočný učiaci faktor stanovený na hodnotu 0,001. Tento faktor sa počas trénovania automaticky upravoval, po dosiahnutí 55000 iterácií klesla jeho hodnota o jedno desatinné miesto, celkový počet iterácií bol 385000. Zo všetkých obrázkov pre trénovanie boli vyfiltrované obrázky s veľmi malými alebo naopak veľmi veľkými rozmermi a ich rozdelenie bolo zvolené v pomere 50:50, čo znamená 7704 obrázkov pre trénovanie a 7693 obrázkov pre testovanie. V tomto prípade môžeme hovoriť o celkovom počte 50 epoch. Celkový čas trénovania bol 15 hodín a 39 minút. Vývoj stratovej funkcie pri trénovaní na GPU 1 je možné vidieť na obr. 6.9.
`GPU 0`: Pre trénovanie GPU 0 bolo zachované rovnaké nastavenie parametrov ako v prípade trénovania prostredníctvom GPU 1. Celkový čas trénovania bol 8 hodín a 25 minút. Vývoj stratovej funkcie pri trénovaní na GPU 0 je možné vidieť na obr. 6.10. Na základe testovacích dát bola vyhodnotená presnosť trénovaných modelov. Porovnanie dosiahnutej presnosti na oboch dostupných GPU je zobrazené na obr. 6.11. Obsahom tabuľky 6.5 je prehľad vyššie zmienených parametrov a dosiahnutých presností trénovaných modelov s využitím oboch grafických kariet. Ukážka výsledku segmentácie prostredníctvom modelu s najlepšou presnosťou je možné vidieť na obr. 6.12.



Obr. 6.9: Graf trénovania modelu pre sémanticku segmentáciu prostredníctvom GPU NVIDIA GEFORCE GTX 980 a na základe datasetu Fashion



Obr. 6.10: Graf trénovania modelu pre sémanticku segmentáciu prostredníctvom GPU NVIDIA GEFORCE GTX 1080 Ti a na základe datasetu Fashion



Obr. 6.11: Graf vývoja presnosti počas testovania natrénovaných modelov pre obe GPU. Zdrojom obrazových dát bol dataset Fashion

Tab. 6.5: Prehľad zvolených parametroch pre trénovanie a nadobudnutá presnosť na testovacích dátach pomocou datasetu Fashion. Veľkosť učiaceho faktoru platí iba do momentu dosiahnutia počtu iterácií uvedených v odpovedajúcom stĺpci

Grafická karta NVIDIA GTX 980							
Počet iterácií	55000	110000	165000	220000	275000	330000	385000
Učiaci faktor	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}
Presnosť IoU [%]	46,84	49,80	50,19	50,42	50,43	50,42	50,42
Grafická karta NVIDIA GTX 1080 Ti							
Počet iterácií	55000	110000	165000	220000	275000	330000	385000
Učiaci faktor	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}
Presnosť IoU [%]	47,82	49,35	50,71	50,73	50,74	50,72	50,73

Obr. 6.12: Príklad realizácie segmentácie na natrénovanom modeli. V ľavej časti sa nachádza originálny obrázok a v pravej časti výsledný obrázok po segmentácii [47]

7 DISKUSIA VÝSLEDKOV

Kapitola sa venuje diskusii dosiahnutých výsledkov v tejto práci.

V rámci teoretickej časti boli najskôr preštudované a spísané princípy existujúcich prístupov v oblasti počítačového videnia a základné prvky neurónových sietí až po konvolučné neurónové siete. Zmienená teoretická časť práce je tak sústredená v kapitole 1 a 2.

Pre otestovanie funkčnosti segmentácie bola použitá metóda CRF-RNN kedy sa využíva princíp trénovania konvolučných neurónových sietí a podmienených náhodných polí celou sieťou. Základná činnosť tejto metódy je bližšie opísaná v kapitole 3. V ďalšej časti, pri snahe trénovania vlastného modelu pomocou tejto metódy neboli dosiahnuté očakávané výsledky v konečnom segmentovanom obrázku. Tento problém bol prípsaný nesprávnej inicializácii dekonvolučnej vrsty, ktorú sa aj napriek vynaloženej snahe nepodarilo v práci správne iniciovať. Okrem zmienených neuspokojivých výsledkov je metóda limitujúca i z pohľadu kapacity potrebnej pre jej realizáciu. Nakoľko mal systém v čase trénovania k dispozícii GPU NVIDIA GTX 980 (GPU 1) s menšou kapacitou RAM ako so všeobecnými výpočtovými požiadavkami pre realizáciu segmentácie danou metódou, rozmery obrázkov bolo preto nutné zmenšiť. Z dôvodu týchto limitácií bola naštudovaná a zvolená iná metóda založená na podobnom princípe, kedy je hlavnou odlišnosťou využitie podmienených náhodných polí ako dodatočného spracovania výstupného obrázku. Táto metóda, s názvom Deeplab, odstránila všetky predchádzajúce limitácie a jej aplikovaním bolo možné trénovanie i na základe obrázkov s väčšími rozmermi. Teoretický popis metódy je taktiež zahrnutý v rámci kapitoly 3.

Prierez niektorými prostrediami pre hlboké učenie sa nachádza v kapitole 4 a obsahuje taktiež zvlášť popis konkrétneho vybraného prostredia, pomocou ktorého bol vytvorený systém pre trénovanie a testovanie modelov za účeľom segmentácie. Prvým výsledkom je porovnanie výstupných obrázkov segmentácie predtrénovaného modelu na základe metódy CRF-RNN s výstupnými obrázkami segmentácie vlastným natrénovaným modelom metódou Deeplab. Zmienené výsledky je možné nájsť v rámci príloh k textu práce A.3. Trénovanie malo byť v rámci práce založené na vlastnej databáze. Databáza je vo väčšine prípadov segmentácie obrazu rozčlenená na originálne obrázky, ktoré slúžia ako vstup, pričom nie sú žiadnym spôsobom upravené a na ground-truth obrázky, na ktorých sú anotované objekty, ktoré chceme segmentáciou rozpoznať do tried. Tvorba takejto predlohy je však neúmerne časovo náročná a vytvorenie dostatočnej databázy nie je možné v rámci prideleného času na realizáciu práce. Z toho dôvodu boli zdrojom obrazových dát pre vlastné trénovanie zvolené dve voľne dostupné databázy s odlišným zameraním. Dataset VOC 2012 s počtom 2895 ground-truth obrázkov, ktoré boli rozdelené v pomere 50:50

na trénovacie a testovacie. Vo výsledku sú potom pridelené pixely do 20 možných tried ako napríklad pes, mačka, stôl atď. Dataset Fashion bol prispôsobený na účel vlastného trénovania zredukovaním o obrázky, ktoré boli príliš veľkých alebo naopak príliš malých rozmerov v porovnaní s väčšinou obrázkov v datasete. V prípade originálnych obrázkov sa v datasete nachádzali niektoré obrázky v odtieňoch šedej, ktoré by boli nevyhovujúce z pohľadu testovania modelu. Takto upravený dataset obsahoval celkovo 15397 obrazových dát, ktoré boli opäť rozdelené v pomere 50:50.

Podstatu práce tvorí navrhnuté webové rozhranie, ktoré užívateľovi značne zjednodušuje realizáciu sémantickej segmentácie pomocou natrénovaných modelov. Architektúra, vzhľad a základný princíp rozhrania je uvedený v kapitole 5. Webové rozhranie bolo implementované ako aplikácia v jazyku Python, čím bol dosiahnutý súlad so súbormi vybraného prostredia pre hlboké učenie BVLC / Caffe. Vzhľad aplikácie bol uspôsobený pre možnosť využitia najlepších modelov, ktoré vznikli vlastným trénovaním na základe oboch datasetov. V rámci práce nebolo aplikované riešenie s možnosťou trénovania prostredníctvom navrhnutej aplikácie, nakoľko podstatou úspešného trénovania je dôkladná príprava zložená z niekoľkých úkonov, ktoré zaberú nemalý čas. Je to napríklad príprava obrazových dát a ich ďalšia konverzia, príprava ostatných dát či samotné trénovanie, ktoré dosahuje celkový čas vyjadrený v hodinách až dňoch. Z toho dôvodu nie je príliš nápomocné celý proces automatizovať.

Vlastné trénovanie bolo realizované na základe obrazových dát z vyššie zmienených datasetov. Detaily o systéme spolu s výsledkami trénovania a testovania obsahuje kapitola 6. Pred realizáciou vlastného trénovania bol otestovaný inicializačný model na testovacích dátach VOC i dátach Fashion. V prípade dát VOC dosahoval presnosť 32,54 % a v prípade dát Fashion 7,18 %. Pre prvotné trénovanie bola použitá grafická karta GPU 1. Počas vypracovávania práce bola doplnená výkonnejšia grafická jednotka a trénovanie zahájené i na tejto novej grafickej karte NVIDIA GTX 1080 Ti (GPU 0). So zámerom optimalizácie dosiahnutých výsledkov sa jednotlivé parametre upravovali, používali sa rôzne počty iterácií, tempo učenia a pod. Napríklad najvyšší počet iterácií bol 1198200, kedy bolo trénovanie založené na základe obrazových dát z datasetu Fashion, pri rozdelení 70:30, v zmysle trénovacia : testovacia množina. Počet obrázkov na trénovanie bol 11982 a celkovo bolo tak aplikovaných 100 epoch. Faktor učenia bol postupne zmenšovaný každých 100000 iterácií, celkový čas trénovania modelu trvalo 1 deň 2 hodiny a 46 minút pri použití GPU 0. Trénovanie s vyšším počtom iterácií však nemalo efekt dosiahnutia vyššej presnosti v porovnaní s najlepším nadobudnutým výsledkom. Presnosť sa od určitého počtu iterácií už ďalej nezvýšila. Nastavenie parametrov siete pri ktorom bola získaná najvyššia presnosť je opísané v kap. 6.

Finálna presnosť trénovaného modelu na GPU 1 dosahuje 64 % pri vstupných

dátach zo sady VOC a 50,42 % pri vstupných dátach zo sady Fashion. Dosiahnutá presnosť v prípade využitia GPU 0 a datasetu VOC narástla na 68,52 %. V prípade datasetu Fashion sa presnosť pohybovala v takmer rovnakých hodnotách ako v prípade menej výkonnej GPU 1 a najlepším výsledkom bola presnosť 50,74 %. Výsledné hodnoty závislosti presnosti na počte iterácií boli aproximované a zobrazené v grafoch obr. 6.7 a 6.11. Porovnaním výsledkov nadobudnutých trénovaním a testovaním na oboch GPU je možné tvrdiť, že trénovanie GPU 0 je takmer o polovicu rýchlejšie ako trénovanie na GPU 1. Vplyv na rozdiel medzi výslednou presnosťou oboch GPU v prípade datasetu VOC mohla mať náhodná inicializácia váh, vyššia pamäť v prípade GPU 0 či vhodnejšie zvolený faktor učenia pri danej povahe obrazových dát. Výsledky segmentácie modelu VOC a Fashion je možné nájsť i v prílohe A.3 a A.4.

S chybným priradením pixelov do nesprávnej triedy je možné sa stretnúť najčastejšie pri triedach, ktoré sú si veľmi blízke, napríklad triedy sukňa a šaty alebo často v prípade splývajúceho objektu s pozadím kedy pixely nie sú priradené žiadnej konkrétnej triede a sú tak označené indexom 0 pre pozadie. Príklad chybného priradenia na úrovni pixelov je možné vidieť na obr. A.7 v rámci prílohy A.5.

Využitím metódy Deeplab a vybraných datasetov s príslušným počtom obrázkov na trénovanie je možné považovať ziskané výsledky testovania vlastných natrénovaných modelov za adekvátne. Ďalšou optimalizáciou parametrov, upravením pomeru pre rozdelenie dát či navýšením výpočtových kapacít je možné docieliť vyššej presnosti u budúcich trénovaných modelov.

8 ZÁVER

Práca sa zaoberala problematikou segmentácie obrazu s využitím hlbokého učenia, neurónovými sieťami, ich popisu a vlastnosťami. Na niekoľkých obrázkoch je poukázané na architektúru či princíp fungovania. Zhrnutím prvej časti, v ktorej sú opisované už existujúce a stále sa zdokonalujúce formy využitia neurónových sietí v oblasti hlbokého učenia možno tvrdiť, že vývojári a vedci zaoberajúci sa touto tématikou prichádzajú so stále inovatívnejšími a progresívnejšími metódami pre zlepšenie výkonu a možnosti ich univerzálnejšieho využitia v rôznych oblastiach. Dá sa taktiež predpokladať s ešte väčšou podporu od vznikajúcich výskumných centier a univerzít pre čo najlepšie využitie potenciálu hlbokého učenia a neurónových sietí.

Z faktov uvedených v práci možno zhrnúť, že účeľom hlbokého učenia je istý druh učenia umelej inteligencie zo skúsenosti a porozumenia konkrétneho problému bez potreby presne definovaných inštrukcií a vytvorených príkazov. V značnej väčšine prípadov sa využívajú konvolučné neurónové siete s veľkou hĺbkou, ktorých realizácia je síce výpočetne náročná avšak vďaka súčasnému napredovaniu techník a kapacity výpočtových systémov je tento spôsob prevedenia prístupnejší. Za zámerom úspešnej implementácie konkrétnej metódy s využitím hlbokého učenia boli vyvinuté rôzne prostredia, v práci sa využíva BVLC / Caffe v integrácii s rozhraním jazyka Python spolu s ďalšími súčasťami ako je CUDA od spoločnosti NVIDIA podporujúce lepší výkon.

Hlavným prínosom práce je vytvorenie webového rozhrania pre segmentáciu obrazu s využitím hlbokého učenia. Webové rozhranie bolo implementované ako webová aplikácia v jazyku Python a bolo navrhnuté tak aby umožňovalo výber obrázku určeného pre segmentáciu, jeho spracovanie zvolenou metódou a zobrazenie výsledného obrázku po úspešnej segmentácii trénovanými modelmi. Metóda, ktorá realizuje segmentáciu obrazu je založená na princípe konvolučných neurónových sietí s dodatočným spracovaním pomocou podmienených náhodných polí. Úspešným overením funkčnosti tejto metódy bolo natrénovanie vlastných modelov, ktoré sa využívajú pre riešenie pixelovo orientovaných problémov s následným priradením pixelov do zodpovedajúceho počtu definovaných tried. Dosiahnuté presnosti sú potom: 68,52 % v prípade trénovania s využitím datasetu VOC a 50,74 % v prípade trénovania s využitím datasetu Fashion. Obe zmienené presnosti boli získané na grafickej karte NVIDIA GEFORCE GTX 1080 Ti.

Výhodou navrhnutej aplikácie je, že rozhranie je možné uplatniť pre akýkoľvek model založený na segmentácii obrazu pomocou prostredia hlbokého učenia BVLC / Caffe. Aplikácia môže byť v budúcnosti rozšírená o inú metódu pri zachovaní rovnakej verzie prostredia či ďalej prispôsobená jednoduchou úpravou vzhľadu pre výsledné obrazové dáta s odlišným zameraním.

LITERATÚRA

[1] IMAGE-NET.ORG: *ImageNet Large Scale Visual Recognition Challenge 2016* [online]. 2016 [cit. 1. 11. 2016]. Dostupné z URL: <http://image-net.org/challenges/LSVRC/2016>.

[2] REN, S.; HE, K.; GIRSHICK, R. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks* [online]. 6. 6. 2016 [cit. 1. 11. 2016]. Dostupné z URL: <http://ieeexplore.ieee.org/document/7485869>.

[3] HONG, S.; YOU, T.; KWAK, S.; HAN, B. *Online Tracking by Learning Discriminative Saliency Map with Convolutional Neural Network* [online]. 24. 2. 2016 [cit. 2. 11. 2016]. Dostupné z URL: <https://arxiv.org/pdf/1502.06796.pdf>.

[4] DONG, CH.; HE, K.; LOY, CH.; TANG, X. *Image Super-Resolution Using Deep Convolutional Networks* [online]. 1. 2. 2016 [cit. 2. 11. 2016]. Dostupné z URL: <http://ieeexplore.ieee.org/document/7115171>.

[5] XIE, S.; ZHUOWEN, T. *Holistically-Nested Edge Detection* [online]. Computer Vision (ICCV), 2015 IEEE International Conference 7–13 12. 2015 [cit. 2. 11. 2016]. Dostupné z URL: <http://ieeexplore.ieee.org/document/7410521>.

[6] ARBELÁEZ, P; et al. *Semantic Segmentation using Regions and Parts* [online]. 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [cit. 22. 11. 2016]. Dostupné z URL: <http://people.eecs.berkeley.edu/~arbelaez/Publications.html>.

[7] MOSTAJABI, M.; YADOLLAHPOUR, P.; SHAKNAROVICH, G. *Feedforward semantic segmentation with zoom-out features* [online]. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [cit. 22. 11. 2016]. Dostupné z URL: <https://arxiv.org/pdf/1412.0774.pdf>.

[8] EIGEN, D.; PUHRSCH, CH.; FERGUS, R. *Depth Map Prediction from a Single Image using a Multi-Scale Deep Network* [online]. 2014, [cit. 22. 11. 2016]. Dostupné z URL: <https://www.cs.nyu.edu/~deigen/depth>.

[9] LONG, J.; SHELHAMER, E.; DARRELL, T. *Fully convolutional networks for semantic segmentation* [online]. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [cit. 22. 11. 2016]. Dostupné z URL: <http://ieeexplore.ieee.org/document/7298965>.

[10] BELL, S. et al. *Material recognition in the wild with the Materials in Context Database* [online]. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [cit. 22. 11. 2016]. Dostupné z URL: <http://ieeexplore.ieee.org/document/7298970/>.

[11] CHEN, L-C. et al. *Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs* [online]. 2014, posledná aktualizácia 7. 6. 2016 [cit. 22. 11. 2016]. Dostupné z URL: <https://arxiv.org/abs/1412.7062>.

[12] KENDALL, A. *SegNet* [online]. 2015, [cit. 3. 11. 2016]. Dostupné z URL: <http://mi.eng.cam.ac.uk/projects/segnet>.

[13] KENDALL, A.; BADRINARYANAN, V.; CIPOLLA, R. *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation* [online]. 2015, [cit. 3. 11. 2016]. Dostupné z URL: <http://mi.eng.cam.ac.uk/projects/segnet>.

[14] VARSHNEY, V.; SINGH, D.; TIWARI, A. *Deep learning and its application in silent sound technology* [online]. Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference 16–18. 3. 2016 [cit. 5. 11. 2016]. Dostupné z URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7724756>.

[15] DENG, L.; HINTON, G.; KINGSBURY, B. *New types of deep neural network learning for speech recognition and related applications: an overview* [online]. Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference 26–31. 3. 2013 [cit. 5. 11. 2016]. Dostupné z URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6639344>.

[16] UCAR, A.; DEMIR, Y.; GUZELIS, C. *Moving towards in object recognition with deep learning for autonomous driving applications* [online]. INnovations in Intelligent SysTems and Applications (INISTA), 2016 International Symposium 2–5. 8. 2013 [cit. 5. 11. 2016]. Dostupné z URL: <http://ieeexplore.ieee.org/document/7571862/>.

[17] DIREKOGLUB, C.; MELIKE, S. *Review of MRI-based Brain Tumor Image Segmentation Using Deep Learning Methods* [online]. 12th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS 2016 Vienna, Austria 29–30. 8. 2016 [cit. 5. 11. 2016]. Dostupné z URL: <http://www.sciencedirect.com/science/article/pii/S187705091632587X>.

[18] ZHENG, S. et al. *Conditional Random Fields as Recurrent Neural Networks* [online]. 2015, [cit. 3. 11. 2016]. Dostupné z URL: <http://www.robots.ox.ac.uk/~szheng/CRFasRNN.html>.

[19] VA-ST: *SmartSpecs* [online]. 2015, [cit. 5. 11. 2016]. Dostupné z URL: <http://www.va-st.com/smart-specs/>.

[20] ŠÍMA, J.; NERUDA, R. *Teoretické otázky neuronových sítí* Vyd. 1. Praha: Matfyzpress, 1996. 390s.[cit. 24. 10. 2016] ISBN 80-85863-18-9.

[21] ROJAS, R. *Neural networks: A systematic introduction* [online]. 1996, [cit. 26. 10. 2016]. Dostupné z URL: <http://page.mi.fu-berlin.de/rojas/neural/neuron.pdf>.

[22] SINČÁK, P.; ANDREJKOVÁ, G. *Neurónové siete Inžiniersky prístup (1. diel)* [online]. Technická Univerzita Košice a Univerzita P.J. Šafárika, Košice 1996 [cit. 26. 10. 2016]. Dostupné z URL: <http://neuron-ai.tuke.sk/cig/source/publications/books/NS1/html>.

[23] NIELSEN, M. *Neural Networks and Deep Learning* [online]. posledná aktualizácia 22. 1. 2016 [cit. 27. 10. 2016]. Dostupné z URL: <http://neuralnetworksanddeeplearning.com/index.html>.

[24] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning* [online]. Book in preparation for MIT Press 2016 [cit. 31. 10. 2016]. Dostupné z URL: <http://www.deeplearningbook.org>.

[25] KARPATHY, A.; et. al. *CS231n: Convolutional Neural Networks for Visual Recognition* [online]. 2015 [cit. 31. 10. 2016]. Dostupné z URL: <http://cs231n.stanford.edu/syllabus.html>.

[26] KRIZHEVSKY, A; SUTSKEVER, I.; HINTON, G.E. *ImageNet Classification with Deep Convolutional Neural Networks* [online]. 2012 [cit. 31. 10. 2016]. Dostupné z URL: <http://www.cs.toronto.edu/~kriz>.

[27] HE, K.; ZHANG, X.; REN, S.; SUN, J. *Deep Residual Learning for Image Recognition* [online]. 10. 12. 2015 [cit. 31. 10. 2016]. Dostupné z URL: <https://arxiv.org/abs/1512.03385>.

[28] GIRSHICK, R. et. al. *Rich feature hierarchies for accurate object detection and semantic segmentation* [online]. 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [cit. 15. 11. 2016]. Dostupné z URL: <people.eecs.berkeley.edu/~rbg/papers/r-cnn-cvpr.pdf>.

[29] ZHENG, S. et al. *Neural Networks and Deep Learning* [online]. 2016 European Conference on Computer Vision [cit. 19. 11. 2016]. Dostupné z URL: <http://www.robots.ox.ac.uk/~szheng/Res_CRFRNN/ECCV2016_Tutorial_CNN_CRF.pdf>.

[30] KRAHENBUHL, P.; KOLTUN, V. *Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials* [online]. 1. 12. 2011 [cit. 21. 11. 2016]. Dostupné z URL: <http://www.philkr.net/papers/2011-12-01-nips/2011-12-01-nips.pdf>.

[31] MALLAT, S. *A Wavelet Tour of Signal Processing, 2nd Edition* Academic Press, 1999. 629s.[cit. 11. 4. 2017] ISBN 0-12-466606-X.

[32] SIMONYAN, K.; ZISSERMAN, A. *Very deep convolutional networks for large-scale image recognition* [online]. 10. 4. 2015 [cit. 29. 3. 2017]. Dostupné z URL: <https://arxiv.org/pdf/1409.1556.pdf>.

[33] YANGQING, J.; et. al. *Caffe: Convolutional Architecture for Fast Feature Embedding* [online]. 2014 [cit. 27. 11. 2016]. Dostupné z URL: <http://ucb-icsi-vision-group.github.io/caffe-paper/caffe.pdf>.

[34] MICROSOFT: *The Microsoft Cognitive Toolkit* [online]. 2016 [cit. 27. 11. 2016]. Dostupné z URL: <https://www.microsoft.com/en-us/research/product/cognitive-toolkit>.

[35] TENSORFLOW: *TensorFlow is an Open Source Software Library for Machine Intelligence* [online]. 2016 [cit. 27. 11. 2016]. Dostupné z URL: <https://www.tensorflow.org>.

[36] THEANO: *Theano: A Python framework for fast computation of mathematical expressions* [online]. 2016, posledná aktualizácia 10. 11. 2016 [cit. 27. 11. 2016]. Dostupné z URL: <http://deeplearning.net/software/theano>.

[37] TORCH: *Torch: A scientific computing framework for LuaJIT* [online]. 2016 [cit. 27. 11. 2016]. Dostupné z URL: <http://torch.ch>.

[38] MXNET: *Flexible and Efficient Library for Deep Learning* [online]. 2016 [cit. 27. 11. 2016]. Dostupné z URL: <http://mxnet.io>.

[39] CHAINER: *A Powerful, Flexible, and Intuitive Framework for Neural Networks* [online]. 2015 [cit. 27. 11. 2016]. Dostupné z URL: <http://chainer.org/>.

[40] KERAS: *Keras: Deep Learning library for Theano and TensorFlow* [online]. 2016 [cit. 27. 11. 2016]. Dostupné z URL: <https://keras.io>.

[41] DEEPLEARNING4J: *Deep Learning for Java: Open-Source, Distributed, Deep Learning Library for the JVM* [online]. 2016 [cit. 27. 11. 2016]. Dostupné z URL: <https://deeplearning4j.org>.

[42] ZHENG, S. *CRF-RNN for Semantic Image Segmentation* [online]. 2016 [cit. 1. 12. 2016]. Dostupné z URL: <https://github.com/torrvision/crfasrnn>.

[43] CHEN, L-C. et al. *Deeplab-public repository* [online]. 2014, posledná aktualizácia 2. 2. 2017 [cit. 20. 3. 2017]. Dostupné z URL: <https://bitbucket.org/deeplab/deeplab-public/src>.

[44] YANGQING, J.; SHELHAMER, E. *Caffe Installation* [online]. 2016 [cit. 1. 12. 2016]. Dostupné z URL: <http://caffe.berkeleyvision.org/installation.htm>.

[45] EVERINGHAM, M.; VAN GOOL, L.; et. al. *Visual Object Classes Challenge 2012 (VOC2012)* [online]. 2016 [cit. 3. 12. 2016]. Dostupné z URL: <http://host.robots.ox.ac.uk/pascal/VOC/voc2012>.

[46] KERSNER, M. *train-Deeplab* [online]. 2015, posledná aktualizácia 19. 2. 2017 [cit. 29. 3. 2017]. Dostupné z URL: <https://github.com/martinkersner/train-DeepLab>.

[47] XIAODAN, L. et al. *Human Parsing with Contextualized Convolutional Neural Network* [online]. 2. 3. 2016 [cit. 1. 4. 2017]. Dostupné z URL: <http://ieeexplore.ieee.org/document/7423822/>.

[48] Obrázok *MEET THE COWS (Holstein)* [online]. [cit. 4. 4. 2017]. Dostupné z URL: <http://www.floridamilk.com/on-the-farm/meet-the-cows.stml#>.

[49] Obrázok *Arabian Brown by TAYSER* [online]. [cit. 4. 4. 2017]. Dostupné z URL: <http://photographyblogger.net/25-magnificent-horse-photos/>.

[50] Obrázok *FÖRVARA drawer* [online]. [cit. 4. 4. 2017]. Dostupné z URL: <https://www.pinterest.com/pin/469429961132071208/>.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

BVLC Berkeley Vision and Learning Center

CNN Convolutional Neural Network — Konvolučné neurónové siete

CNTV The Microsoft Cognitive Toolkit

CRF Conditional Random Field — Podmienené náhodne polia

cuDNN NVIDIA CUDA Deep Neural Network library — Knižnica od spoločnosti NVIDIA pre podporu platformy CUDA v oblasti hlbokých neurónových sietí

DCNN Deep Convolutional Neural Network — Hlboké konvolučné neurónové siete

DL4J Deeplearning4j

R-CNN Region-based Convolutional Neural Network — Kombinácia konvolučných neurónových sietí s využitím oblastného návrhu

FCN Fully Convolutional Neural Network — Plne pospájané konvolučné neurónové siete

GPU Graphics Processing Unit — Grafický procesor

IEEE Institute of Electrical and Electronics Engineers — Inštitúcia, ktorá združuje elektroinžinierov a informatikov a zároveň sa podiela na odborných tvorbe periodík a rade kníh v tejto oblasti

ILSVRC Imagenet Large Scale Visual Recognition Challenge — Výzva Image-Netu pre rozpoznanie obrazov ve veľkom merítku

IoU Intersection over Union — Metrika pre hodnotenie presnosti modelu prostredia hlbokého učenia

MRF Markov Random Fields — Markove náhodné polia

PDP Parallel Distributed Processing Group — Názov pre konkrétnu skupinu, ktorá sa zaoberala výskumom v oblasti neurónových sietí

RNN Recurrent Neural Networks — Rekurentné neurónové siete

RPN Region Proposal Network — Siete s oblastným návrhom

SST Silent Sound Technology — Technológia, ktorá spočíva v prenose informácií bez použitia hlasiviek

SVM Support Vector Machine — Systém podporných vektorov

XOR Exclusive OR — Logická operácia (vylučujúca disjunkcia)

b — bias

E — energia pre priradenie tried

f — primitívna funkcia

f_θ — výsledok jednej iterácie metódou stredného poľa

i — pixel

I — obrázok

k — Gaussový kernel

l — preddefinovaná hodnota tried

M — Gaussový filter

n — premenná pre označenie vstupu

N — počet pixelov v obrázku

o — vstupný kanál neurónu

P — prah

Q — marginálna distribúcia

T — počet iterácií

U_i — negatívna unárna energia

w_o — váha vstupného kanálu

x_o — informácia vstupného kanálu

X_i — priradená trieda pixelu

y — premenná pre označenie výstupu

Z — podielová funkcia

μ — funkcia zosúladenia

ψ_u — energia jednočlennej zložky

ψ_p — energia párových zložiek

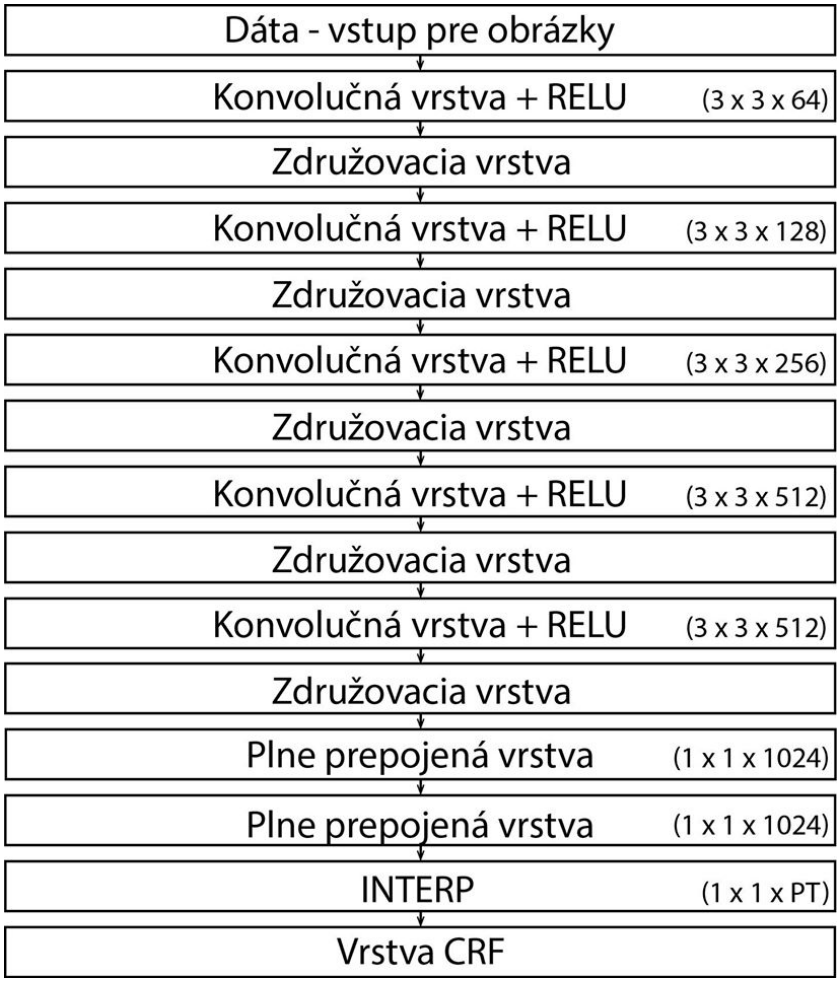
σ — sigmoidná funkcia

θ — vektor pre parametre iterácie

ZOZNAM PRÍLOH

A PRÍLOHY K TEXTU PRÁCE

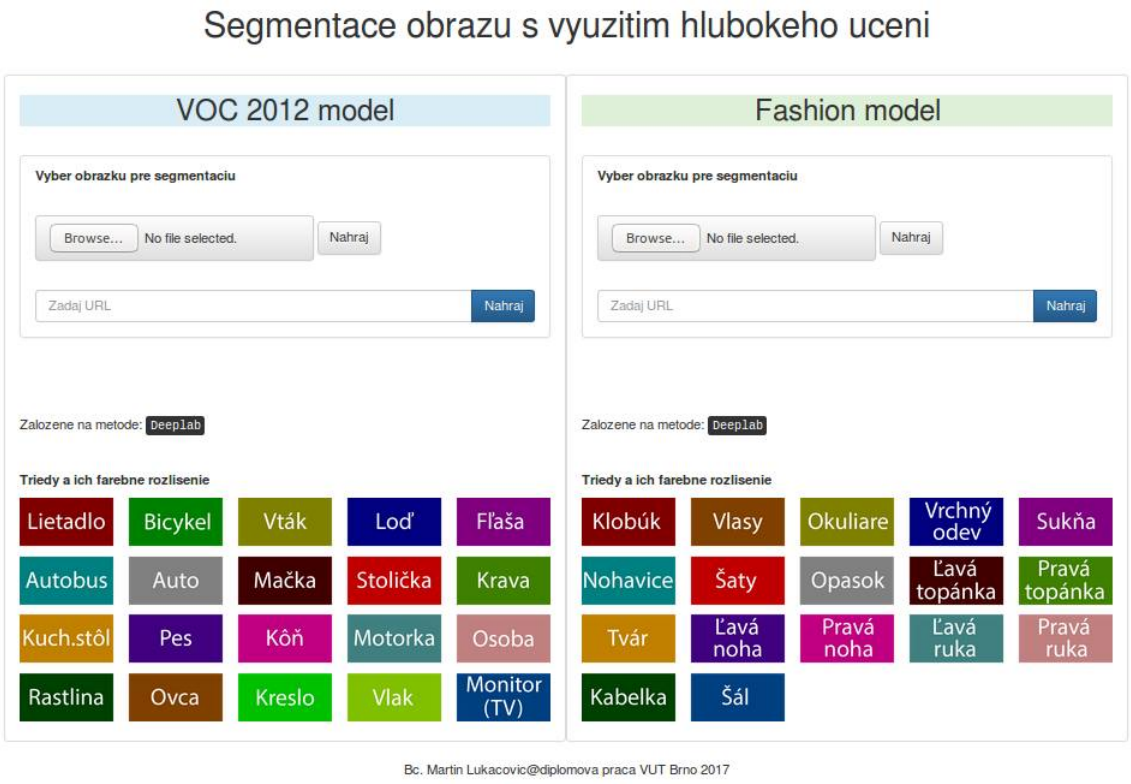
A.1 Diagram vzniknutého modelu pre hlbokú konvolučnú neurónovú sieť



Obr. A.1: Čísla u Konvolučnej vrstvy (Y x Y x Z), odpovedajú rozmerom daných filtrov (Y x Y) a ich počtu (Z). Vo výstupnej vrstve potom PT značí počet tried

A.2 Vzhľad stránky pre webové rozhranie

A.2.1 Úvodná stránka pre výber obrázka



Obr. A.2: Vzhľad úvodnej stránky webového rozhrania

A.2.2 Stránka s vybraným obrázkom a možnosťou segmentácie

Obr. A.3: Vzhľad stránky s vybraným obrázkom, kedy má užívateľ možnosť vrátiť sa na úvodnú stránku alebo spustiť segmentovanie

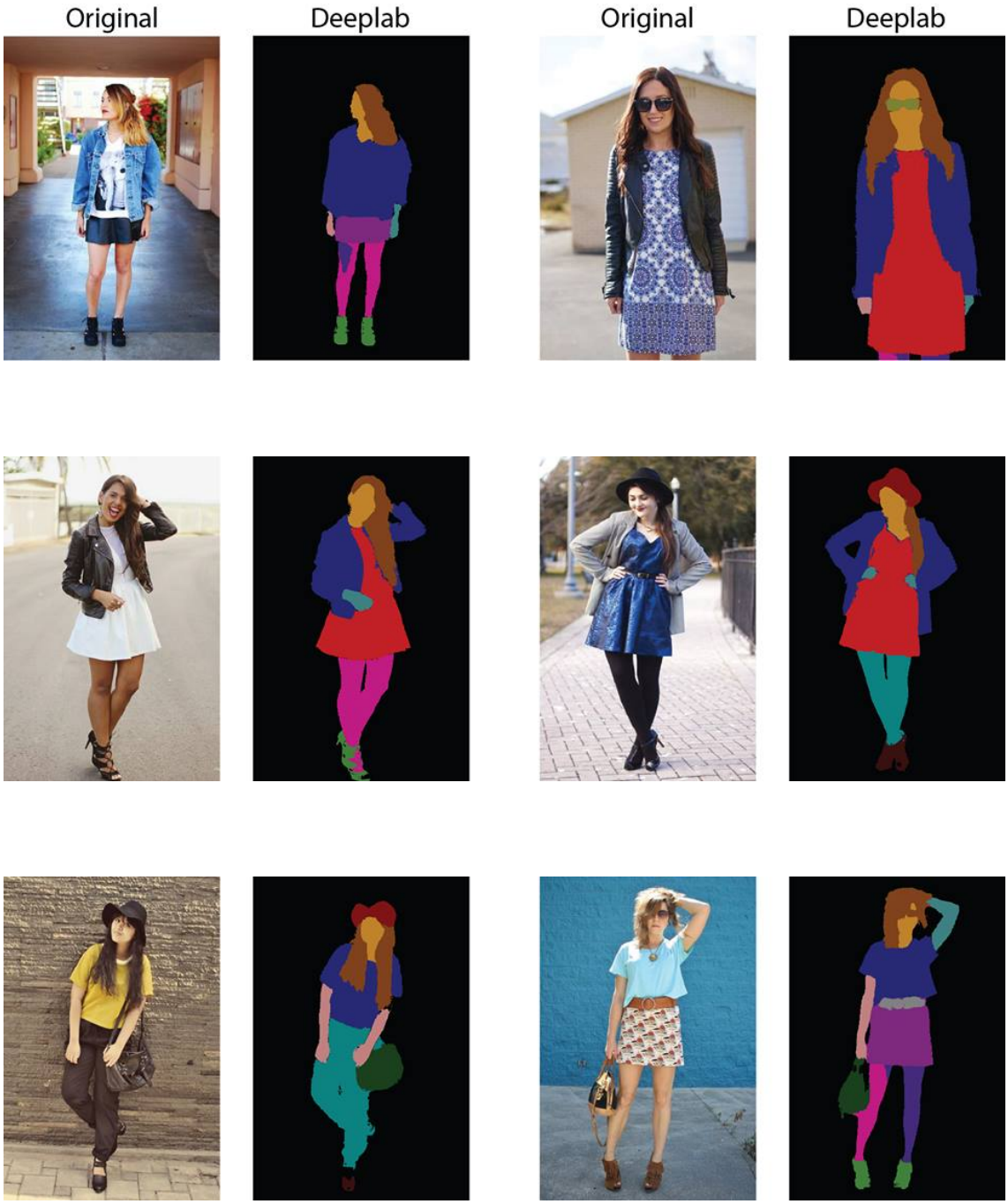
A.2.3 Stránka s výsledkom po segmentácii

Obr. A.4: Vzhľad stránky po úspešnej segmentácii vybraného obrázka

A.3 Výsledky segmentácie – dataset VOC

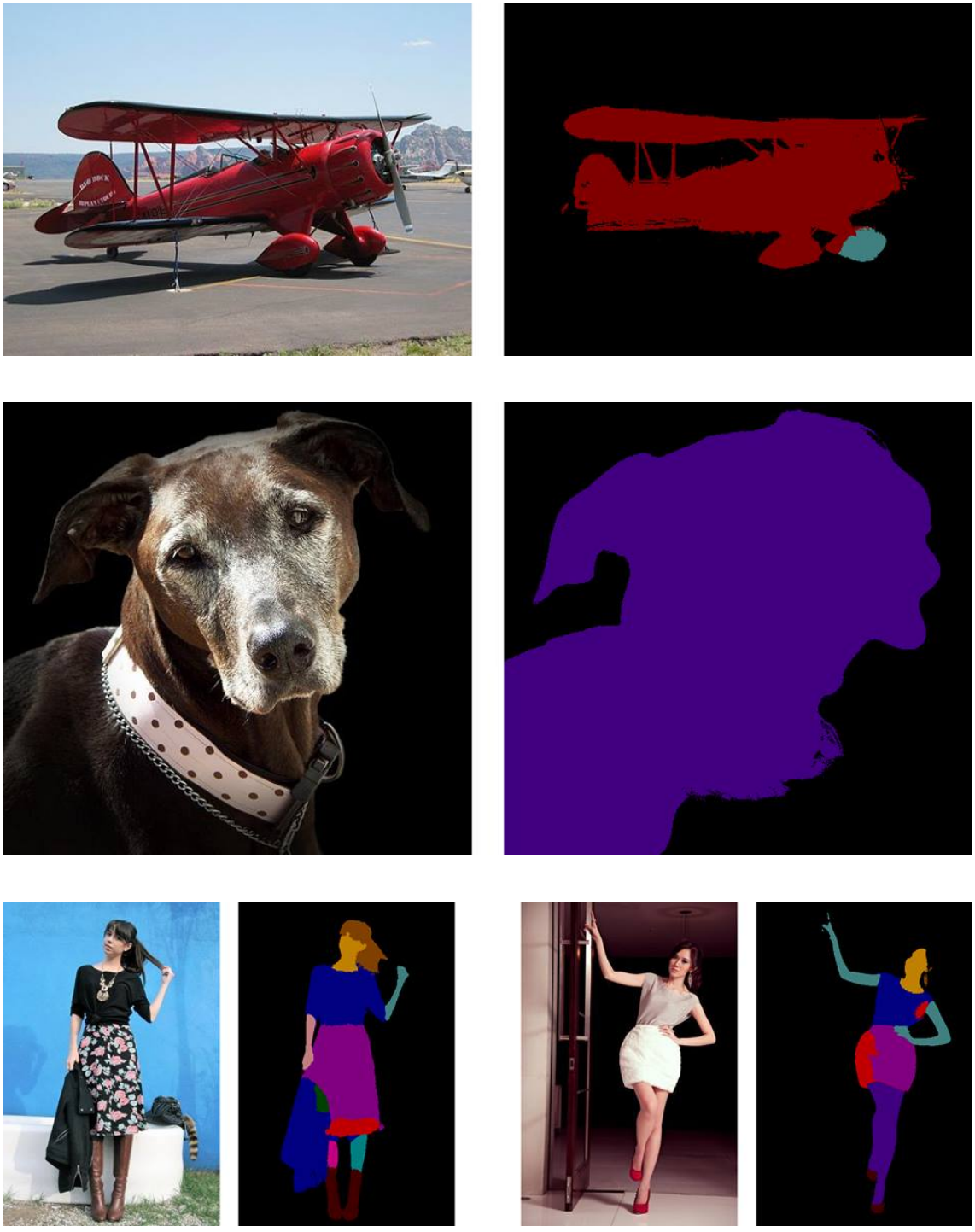
Obr. A.5: Porovnanie výsledkov segmentácie na vybraných obrázkoch. V ľavej časti sú zobrazené originálne obrázky, v strednej časti sú výsledky segmentácie na základe predtrénovaného modelu metódou CRF-RNN a v pravej časti sú výsledky segmentácie na základe vlastného natrénovaného modelu metódou Deeplab. Obrázky sú prevzaté z [48] [49] [50]

A.4 Výsledky segmentácie – dataset Fashion



Obr. A.6: Ukážky segmentácie oblečenia na základe natrénovaného modelu s najvyššou dosiahnutou presnosťou. Každá dvojica začína zľava originálnym obrázkom a k nemu priradenému výsledku po segmentácii s rozpoznanými časťami podľa tried Fashion [47]

A.5 Výsledky segmentácie



Obr. A.7: Ukážky segmentovaných obrázkov, kedy boli pixely nesprávne priradené inej triede, resp. pozadiu [45] [47]

B OBSAH DVD PRILOŽENÉHO K DIPLOMOVEJ PRÁCI

Obsah priloženého DVD k diplomovej práci sa člení do jednotlivých zložiek. Databázy, prípadne odkaz na ne sú umiestnené v zložke `Data`. Súbory potrebné pre vlastné trénovanie a testovanie pomocou prostredia pre hlboké učenie BVLC / Caffe sa nachádzajú v zložke `Implementacia`. Text vlastnej práce je možné nájsť v zložke `Text`. Poslednou zložkou je `Web-app`, ktorá zahŕňa kompletné riešenie webového rozhrania spolu s oboma natrénovanými modelmi prostredia pri využití oboch datasetov.